Fenchel duality-based algorithms for convex optimization problems with applications in machine learning and image restoration

Dissertation

submitted in partial fulfillment of the requirements for the academic degree doctor rerum naturalium (Dr. rer. nat.)

by

André Heinrich

CHEMNITZ UNIVERSITY OF TECHNOLOGY

Prof. Dr. Dr. h. c. (NUM) Gert Wanka, Adviser

Chemnitz, September 28, 2012

Prof. Dr. Dr. h. c. (NUM) Gert Wanka
Technische Universität Chemnitz
Fakultät für Mathematik
D-09107 Chemnitz

Dipl.-Math. oec. André Heinrich
Technische Universität Chemnitz
Fakultät für Mathematik
D-09107 Chemnitz

Bibliographical description

André Heinrich

Fenchel duality-based algorithms for convex optimization problems with applications in machine learning and image restoration

Dissertation, 163 pages, Chemnitz University of Technology, Department of Mathematics, Chemnitz, 2012

Report

The main contribution of this thesis is the concept of Fenchel duality with a focus on its application in the field of machine learning problems and image restoration tasks. We formulate a general optimization problem for modeling support vector machine tasks and assign a Fenchel dual problem to it, prove weak and strong duality statements as well as necessary and sufficient optimality conditions for that primal-dual pair. In addition, several special instances of the general optimization problem are derived for different choices of loss functions for both the regression and the classification task. The convenience of these approaches is demonstrated by numerically solving several problems.

We formulate a general nonsmooth optimization problem and assign a Fenchel dual problem to it. It is shown that the optimal objective values of the primal and the dual one coincide and that the primal problem has an optimal solution under certain assumptions. The dual problem turns out to be nonsmooth in general and therefore a regularization is performed twice to obtain an approximate dual problem that can be solved efficiently via a fast gradient algorithm. We show how an approximate optimal and feasible primal solution can be constructed by means of some sequences of proximal points closely related to the dual iterates. Furthermore, we show that the solution will indeed converge to the optimal solution of the primal for arbitrarily small accuracy.

Finally, the support vector regression task is obtained to arise as a particular case of the general optimization problem and the theory is specialized to this problem. We calculate several proximal points occurring when using different loss functions as well as for some regularization problems applied in image restoration tasks. Numerical experiments illustrate the applicability of our approach for these types of problems.

Keywords

conjugate duality, constraint qualification, double smoothing, fast gradient algorithm, Fenchel duality, image deblurring, image denoising, machine learning, optimality condition, regularization, support vector machines, weak and strong duality

Acknowledgments

First of all, I am much obliged to my adviser Prof. Dr. Gert Wanka for supporting me during the last years and giving me the opportunity to work on this thesis. I would like to thank him for many useful hints and input on research issues as well as help on all matters.

I am strongly indebted to Dr. Radu Ioan Boţ for his unremitting commitment in providing suggestions and being available for countless enlightening discussions to make this thesis possible. Thank you, Radu, you were the one who got the ball rolling and supported me by demanding to go the extra mile that is required to make things work! Thank you also for being a friend beyond scientific issues!

I would like to thank my office mate Dr. Robert Csetnek for useful hints on the thesis, helpful comments and discussions for years.

I am grateful to Prof. Wanka and his whole research group, Christopher, Nicole, Oleg, Radu, Robert and Sorin for the warm-hearted and friendly atmosphere at the department.

It is my duty to thank the European Social Fund and prudsys AG in Chemnitz for supporting my work on this thesis.

Last but not least, I thank Diana for her love, understanding, encouragement and not least for making sacrifices for the purpose of completing this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Duality theory plays an important role in the wide field of optimization and its application. Many optimization tasks arising from real world problems involve the minimization of a convex objective function possibly restricted to a feasible set that is convex. Such applications include data mining issues like classification and regression tasks or other fields like portfolio optimization problems (see [24, 39, 75]) or image restoration tasks in the sense of deblurring and denoising (cf. [56, 6]). Suitable methods for solving regression and classification problems originate among others from the field of statistical learning theory and lead to optimization problems involving a loss function and possibly a regularization functional, all of them assumed to be convex functions but not necessarily differentiable. The famous support vector machines approach represents a problem class arising in the field of statistical learning theory in order to solve both regression and classification tasks. This approach has been extensively studied initially by V.N. Vapnik in [71] and [70] where Lagrange duality plays the dominant role in order to obtain dual optimization problems which have structures more easy to handle than the original primal optimization problems that aim at modelling the specific regression or classification task, respectively. A comprehensive study of the theory of support vector machines can be found in [63]. Such methods belong to the class of kernel based methods ([64]) that have become, especially in the last decade, a popular approach for learning functions from a given set of labeled data. They have wide fields of applications such as image and text classification (cf. [30, 45]), computational biology ([54]), time series forecasting and credit scoring (see [47, 68]) and value function approximation in the field of approximate dynamic programming (cf. [65, 49, 10]).

The support vector machines approach is in the main focus of this thesis since it is investigated by means of Fenchel duality other than the common approach using Lagrange duality. Moreover, this concept will be the basis for numerical tests verifying the good performance of the support vector approach with respect to regression and classification problems on the one hand and for the verification

of the applicability of the algorithm developed in this thesis on the other hand.

There is a variety of algorithms that numerically solve these optimization problems, each of them more or less designed for special structures of the underlying problem, for example, interior point methods for quadratic convex optimization problems, gradient methods for differentiable problems or subgradient methods for nondifferentiable problems. Compared to other methods, for reasons of good rates of convergence of fast gradient methods often nonsmooth optimization problems are regularized in order to solve them via such gradient methods efficiently. Often duality theory is applied since in many cases the dual optimization problem w. r. t. the corresponding primal one has nice properties which allow to numerically solve it in a more easy way. Concerning the famous support vector approach, for example, it is more convenient to solve a Fenchel dual problem or, equivalently, a Lagrange dual problem which may exhibit differentiable quadratic objective functions with simple box constraints or nonnegativity constraints, depending on the chosen loss function. The solution then can be accessed via some well developed algorithms.

In the general setting we consider in this thesis where the primal problem is not assumed to be smooth, we make use of Fenchel duality in addition to a double smoothing technique for efficiently solving the general optimization problem by applying a fast gradient method. In particular, we use a double smoothing technique (see [36]) to be able to apply the fast gradient method introduced in [50] and solve the dual problem. The smoothing is applied since the general dual problem is a non-smooth optimization problem and the fast gradient approach requires a strongly convex, continuously differentiable objective function with Lipschitz continuous gradient.

An important property when solving an optimization problem by means of its dual and obtaining an optimal primal solution is strong duality, namely the case when the optimal primal objective value and the optimal dual objective values coincides and the dual problem has an optimal solution. In general, strong duality can be ensured by assuming that an appropriate regularity condition is fulfilled ([17, 12, 14, 15, 18]). In that case one can also state optimality conditions for primal-dual pairs of optimization problems.

This thesis is structured as follows. Basically it consists of two parts. In the first part a machine learning problem is considered and treated via Fenchel-type duality which is not the common approach in literature for this type of problems. In a second part an approach for solving a general optimization problem approximately is developed and its applicability is demonstrated for two different kinds of application.

In particular, in **Chapter 2** first some basic notations and results from the field of convex analysis that are used in this thesis are presented. Furthermore, we introduce the optimization problems we will deal with in particular in the sub-

sequent chapters. Section 2.2 introduces these two main optimization problems, where in Subsection 2.2.1, the general optimization problem arising in the context of support vector machines for classification and regression ($\mathrm{P_{SV}}$) is derived from the starting point of the Tikhonov regularization problem and allows for modelling this type of tasks. This primal problem is a more general formulation of the standard optimization problem for support vector tasks in the sense that there is the generalized representer theorem (cf. [62]) underlying the construction of this problem. The generalized representer theorem is a generalization of the well known representer theorem ([74]) and allows the use of regularization terms other than the squared norm of the searched classifier or regression function in the corresponding function space. Another general optimization problem ($\mathrm{P_{gen}}$) is then introduced in Subsection 2.2.2. This problem will be extensively studied in Chapter 4. Finally, in Section 2.3 we shortly introduce an algorithmic scheme that will be used for solving ($\mathrm{P_{gen}}$) via its Fenchel dual problem approximately.

In **Chapter 3** we will investigate duality properties between ($\mathrm{P_{SV}}$) and a Fenchel-type dual problem which will be assigned to it. Therefore, in Section 3.1 we will present weak and strong duality statements and optimality conditions. For a special choice of the regularization term in Section 3.2 we will then derive the particular instances of the primal and dual optimization problems that arise by choosing different loss functions especially designed for the classification task. This is followed by an application of the classification task where we numerically solve a classification task based on high dimensional real world data. In the subsequent Section 3.4 we will calculate the corresponding primal and dual problems for the choice of different loss functions for the regression task and apply the results by solving two different regression tasks numerically. The work done in this chapter is based mainly on [20] and [19].

In **Chapter 4** we develop a new algorithm that solves ($\mathrm{P_{gen}}$), introduced in Subsection 2.2.2, approximately. In particular, we assign a Fenchel dual problem ($\mathrm{D_{gen}}$) to ($\mathrm{P_{gen}}$) and then twice apply a smoothing and solve the doubly smoothed problem by using a fast gradient scheme with rate of convergence for the dual objective value to the optimal dual objective value of $O\left(\frac{1}{\varepsilon}\ln\left(\frac{1}{\varepsilon}\right)\right)$. Moreover it is shown that the same rate of convergence holds for the convergence of the gradient of the single smoothed dual objective function to zero. After having performed the double smoothing and verified the convergence there will be constructed an approximately optimal primal solution by means of sequences generated by the algorithm and the convergence of this solution to the optimal solution of ($\mathrm{P_{gen}}$) is shown.

Having theoretically established the approach for solving ($\mathrm{P_{gen}}$) in Chapter 4, we will apply it in **Chapter 5** first to the problem of image restoration which is to solve an ill-posed problem. It is shown that the double smoothing approach developed in Chapter 4 performs well on this problem class and outperforms other methods like the well known fast iterative shrinkage-thresholding algorithm

(FISTA) ([7]) which is an accelerated version of ISTA ([32]). Second, in order to verify its applicability to problems having the sum of more than three functions in the objective, we solve a support vector regression task based on the toy data set that has already been used in Section 3.5.

The thesis concludes with the theses aiming at summarizing the main results of this work.

Chapter 2

Preliminaries and notions

In this chapter we will introduce the basic mathematical background and theoretical concepts needed in the subsequent chapters. First, in the subsequent section some elementary notions from the field of convex analysis are introduced. In Section 2.2 the optimization problems considered in this thesis are presented. There, we first derive the primal optimization problem that arises in the field of machine learning, namely the support vector machines problem for classification and regression. The presented optimization problem will be extensively studied in Chapter 3 in the view of Fenchel-type duality. In a second subsection therein we state a general optimization problem that will be studied in Chapter 4 also in view of Fenchel-type duality with the focus on an algorithmic scheme which is meant to solve this problem approximately. Finally, the fast gradient method used for numerically solving the smoothed dual problem is presented in the last section of this chapter.

2.1 Convex Analysis

In this section we present notions and preliminaries from convex analysis. It mainly refers to [17] and [5]. For a comprehensive study of the elements of convex analysis we refer also to [58, 40, 77, 37]. In this thesis we restrict ourselves to finite dimensional vector spaces, while the concepts of convex analysis can be found in the mentioned literature in more general settings.

In the whole thesis $\mathbb{R}^n$ will denote the n-dimensional real vector space while $\mathbb{R}^{m\times n}$ denotes the space of real $m \times n$ matrices. The extended real space will be denoted by $\overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$ where we assume by convention (see [17])

$$(+\infty) + (-\infty) = (-\infty) + (+\infty) := +\infty,$$
$$0 \cdot (+\infty) := +\infty \quad \text{and} \quad 0 \cdot (-\infty) := 0.$$

By $e_i \in \mathbb{R}^n$, $i = 1, \ldots, n$, the i-th unit vector of $\mathbb{R}^n$ will be denoted. The nonnegative real numbers are denoted by $\mathbb{R}_+ := [0, +\infty)$. Let $U \subseteq \mathbb{R}^n$ be a

subset of $\mathbb{R}^n$. Then the interior of U will be denoted by $\operatorname{int}(U)$ and the closure of it by $\operatorname{cl}(U)$ whereas the boundary of U is $\operatorname{bd}(U) = \operatorname{cl}(U) \setminus \operatorname{int}(U)$. The affine hull of U is defined by

$$\operatorname{aff}(U) := \left\{ \sum_{i=1}^{n} \lambda_i x_i : n \in \mathbb{N}, x_i \in U, \lambda_i \in \mathbb{R}, i = 1, \ldots, n, \sum_{i=1}^{n} \lambda_i = 1 \right\}$$

while the relative interior of U is defined by

$$\operatorname{ri}(U) := \{x \in \operatorname{aff}(U) : \exists \varepsilon > 0 : B(x, \varepsilon) \cap \operatorname{aff}(U) \subseteq U\},$$

where $B(x, \varepsilon)$ is the open ball centered at $x \in \mathbb{R}^n$ and with radius $\varepsilon > 0$. If U is a convex set and $\operatorname{int}(U) \neq \emptyset$, then $\operatorname{int}(U) = \operatorname{int}(\operatorname{cl}(U))$ and $\operatorname{cl}(\operatorname{int}(U)) = \operatorname{cl}(U)$. The indicator function $\delta_U : \mathbb{R}^n \to \overline{\mathbb{R}}$ of a set $V \subseteq \mathbb{R}^n$ is given by

$$\delta_V(x) := \begin{cases} 0, & \text{if } x \in V, \\ +\infty, & \text{otherwise.} \end{cases}$$

For $x, y \in \mathbb{R}^n$ we denote by $\langle x, y \rangle := \sum_{i=1}^{n} x_i y_i$ the inner product and by $\|x\|$ the euclidean norm. In the following four different convexity notions for real valued functions will be presented (see [17, 5]).

Definition 2.1. A function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is called convex if for all $x, y \in \mathbb{R}^n$ and all $\lambda \in [0, 1]$ it holds

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda) f(y). \tag{2.1}$$

A function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is called concave if $-f$ is convex.

For a convex set $U \subseteq \mathbb{R}^n$ a function $f : U \to \overline{\mathbb{R}}$ is called convex on U if (2.1) holds for all $x, y \in U$ and every $\lambda \in [0, 1]$. Analogously, a function f is said to be concave on U if $-f$ is convex on U. The extension of the function f to the whole space $\overline{\mathbb{R}}$ is the function

$$\tilde{f} : \mathbb{R}^n \to \overline{\mathbb{R}}, \quad \tilde{f}(x) = \begin{cases} f(x), & \text{if } x \in U, \\ +\infty, & \text{otherwise.} \end{cases} \tag{2.2}$$

Besides the usual notion of convexity of a function the concepts of strict convexity and strong convexity will play a role in the following.

Definition 2.2. A function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is called strictly convex if for all $x, y \in \mathbb{R}^n$ with $x \neq y$ and all $\lambda \in (0, 1)$ it holds

$$f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda) f(y). \tag{2.3}$$

A function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is called strictly concave if $-f$ is strictly convex.

Via the more restrictive concept of uniform convexity, a special case of it, namely strong convexity, is introduced. To do that further notions are required. The domain of a function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is $\operatorname{dom} f := \{x \in \mathbb{R}^n : f(x) < +\infty\}$. A function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is called proper if $f(x) > -\infty$ for all $x \in \mathbb{R}^n$ and $\operatorname{dom} f \neq \emptyset$.

Definition 2.3. Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be proper. Then f is called uniformly convex with modulus $\phi : \mathbb{R}_+ \to [0, +\infty]$ if ϕ is increasing, $\phi(x) = 0$ only for $x = 0$ and for all $x, y \in \operatorname{dom} f$ and all $\lambda \in (0, 1)$ it holds

$$f(\lambda x + (1-\lambda)y) + \lambda(1-\lambda)\phi(\|x - y\|) \leq \lambda f(x) + (1-\lambda)f(y) \tag{2.4}$$

Definition 2.4. A proper function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is called strongly convex if relation (2.4) holds with $\phi(\cdot) = \frac{\beta}{2}|\cdot|^2$ for some $\beta > 0$.

For these four different convexity notions it holds the following. A strongly convex function is also uniformly convex. A uniformly convex function is also strictly convex and a strictly convex function is convex, too. In the subsequent considerations, especially when dealing with the Moreau envelop, we will need the infimal convolution of finitely many functions.

Definition 2.5. The infimal convolution of the functions $f_i : \mathbb{R}^n \to \overline{\mathbb{R}}$, $i = 1, \ldots, m$, $m \in \mathbb{N}$, is the function $f_1 \square \ldots \square f_m : \mathbb{R}^n \to \overline{\mathbb{R}}$,

$$(f_1 \square \ldots \square f_m)(x) := \inf_{\substack{x^i \in \mathbb{R}^n, \\ \sum_{i=1}^m x^i = x}} \left\{ \sum_{i=1}^m f_i(x^i) \right\}.$$

The infimal convolution is called exact at x if the infimum is attained for $x \in \mathbb{R}^n$.

In particular, for $m = 2$, denote $f_1 = f$ and $f_2 = g$, the infimal convolution is given by

$$(f \square g)(x) = \inf_{\substack{y,z \in \mathbb{R}^n \\ y+z=x}} \{f(y) + g(z)\} = \inf_{y \in X} \{f(y) + g(x-y)\}.$$

With the help of these notions we introduce an important tool for our analysis in Chapter 4, namely the Moreau envelop of a function (see [5]). We therefore need the notion of lower semicontinuity of a function.

Definition 2.6. A function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is called lower semicontinuous at $\bar{x} \in \mathbb{R}^n$ if $\liminf_{x \to \bar{x}} f(x) \geq f(\bar{x})$. If f is lower semicontinuous at all $x \in \mathbb{R}^n$ it is called lower semicontinuous.

It can be shown that the function f is lower semicontinuous if and only if epi$f$ is closed, where the set epi$f := \{(x, \xi) \in \mathbb{R}^n \times \mathbb{R} : f(x) \leq \xi\}$ is called the epigraph of f.

Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be proper, convex and lower semicontinuous and $\gamma > 0$. The Moreau envelop of the function f of parameter γ is defined as

$$ {}^{\gamma}f(x) := \left(f \square \frac{1}{2\gamma} \| \cdot \|^2 \right)(x) = \inf_{y \in \mathbb{R}^n} \left\{ f(y) + \frac{1}{2\gamma} \|x - y\|^2 \right\}. \tag{2.5} $$

Further we denote by $\text{Prox}_f(x)$ the unique point in $\mathbb{R}^n$ that satisfies

$$ {}^{1}f(x) = \min_{y \in \mathbb{R}^n} \left\{ f(y) + \frac{1}{2}\|x - y\|^2 \right\} = f(\text{Prox}_f(x)) + \frac{1}{2}\|x - \text{Prox}_f(x)\|^2 $$

and the operator $\text{Prox}_f : \mathbb{R}^n \to \mathbb{R}^n$ is called the proximity operator of f. Moreover it holds

$$ {}^{\gamma}f(x) = f(\text{Prox}_{\gamma f}(x)) + \frac{1}{2\gamma}\|x - \text{Prox}_{\gamma f}(x)\|^2, $$

where

$$ \text{Prox}_{\gamma f}(x) = \operatorname*{argmin}_{y \in \mathbb{R}^n} \left\{ f(y) + \frac{1}{2\gamma}\|x - y\|^2 \right\} $$

is the unique solution of the minimization problem occurring in (2.5) and is called the proximal point of x. To verify the existence and uniqueness of a minimizer of this problem we refer to [5, Proposition 12.15]. An important property we get from [38, Satz 6.37]. Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a proper, convex and lower semicontinuous function and ${}^{\gamma}f : \mathbb{R}^n \to \mathbb{R}$ its Moreau envelop of parameter $\gamma > 0$. Then ${}^{\gamma}f$ is continuously differentiable and its gradient is given by

$$ \nabla ({}^{\gamma}f)(x) = \frac{1}{\gamma}(x - \text{Prox}_{\gamma f}(x)) \tag{2.6} $$

for all $x \in \mathbb{R}^n$. A basic concept considering the optimization problems and the formulation of their dual problems studied in this thesis are the concepts of conjugate functions and subdifferentiability. Therefore we next define these notions.

Definition 2.7. The (Fenchel) conjugate function $f^* : \mathbb{R}^n \to \overline{\mathbb{R}}$ of a function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is defined by

$$ f^*(x^*) := \sup_{x \in \mathbb{R}^n} \{\langle x^*, x \rangle - f(x)\}, \tag{2.7} $$

while the conjugate $f_S^* : \mathbb{R}^n \to \overline{\mathbb{R}}$ function of f with respect to the nonempty set $S \subseteq \mathbb{R}^n$ is defined by

$$ f_S^*(x) := (f + \delta_S)^*(x) = \sup_{x \in S} \{\langle x^*, x \rangle - f(x)\}. $$

The conjugate function f^* is convex and lower semicontinuous (see [17, Remark 2.3.1],[37, 77]). For all $x^* \in \mathbb{R}^n$ it holds $f^*(x^*) = \sup_{x \in \operatorname{dom} f}\{\langle x, x^*\rangle - f(x)\}$. Besides the conjugate function we can assign the biconjugate function of f, which is defined as the conjugate function of the conjugate f^*, i. e.

$$f^{**} : \mathbb{R}^n \to \overline{\mathbb{R}}, \qquad f^{**}(x) := (f^*)^*(x) = \sup_{x^* \in \mathbb{R}^n} \{\langle x^*, x\rangle - f^*(x^*)\}\,.$$

A famous result w. r. t. conjugate functions is the Fenchel-Moreau theorem.

Theorem 2.8. *Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a proper function. Then $f = f^{**}$ if and only if f is convex and lower semicontinuous.*

For a function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ and its conjugate function the Young-Fenchel inequality holds, i. e. for all $x, x^* \in \mathbb{R}^n$ we have

$$f(x) + f^*(x^*) \geq \langle x^*, x\rangle\,. \tag{2.8}$$

Subdifferentiability is essential when studying convex nondifferentiable optimization problems as is done in the subsequent chapters. Interpreted as a set-valued operator the subdifferential is a maximally monotone operator and leads to special optimization problems arising from the more general formulation of the problem of finding the zeros of the sums of maximally monotone operators (cf. [13]). Next we define the subdifferential of a function at a certain point (see [58, 17, 37, 77]).

Definition 2.9. Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a given function. Then, for any $x \in \mathbb{R}^n$ with $f(x) \in \mathbb{R}$ the set

$$\partial f(x) := \{x^* \in \mathbb{R}^n : f(y) - f(x) \geq \langle x^*, y - x\rangle \quad \forall y \in \mathbb{R}^n\}$$

is said to be the (convex) subdifferential of f at x. Its elements are called subgradients of f at x. We say that the function f is subdifferentiable at x if $\partial f(x) \neq \emptyset$. If $f(x) \notin \mathbb{R}$ we set by convention $\partial f(x) = \emptyset$.

The set-valued mapping $\partial f : \mathbb{R}^n \to \partial f(x)$ is called the subdifferential operator of f. If a convex function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is differentiable at $x \in \mathbb{R}^n$ then the subdifferential of f at this point coincides with the usual gradient of f at x (cf. [58, Theorem 25.1]). A characterization of the elements $x^* \in \partial f(x)$ can be found in [17, Theorem 2.3.12] (see also [77, 37]) which states that

$$x^* \in \partial f(x) \quad \Leftrightarrow \quad f(x) + f^*(x^*) = \langle x^*, x\rangle\,, \tag{2.9}$$

i. e. $x^* \in \partial f(x)$ if and only if the Young-Fenchel inequality (2.8) is fulfilled with equality.

2.2 Primal optimization problems

In this section we will introduce the primal optimization problems we will deal with in this thesis in Chapters 3 and 4. These problems will be investigated via Fenchel-type duality in the corresponding chapter.

2.2.1 Optimization problems in machine learning

This subsection is dedicated to the derivation of a primal optimization problem occurring in the field of supervised learning methods. These methods are referred to the theory of statistical learning and we will consider especially the support vector machines approach for solving classification and regression tasks.

In the following we will give a brief overview of the concept of reproducing kernel Hilbert spaces in the context of kernel based learning methods. These reproducing kernel Hilbert spaces were first introduced by N. Aronszajn (cf. [2]) in the middle of the last century. A deeper insight into this field can be found for example in [64]. In the context of support vector machines for classification and regression the kernel approach allows finding nonlinear classification or regression functions, respectively, while in its original formulation only linear separation of, for example, different classes of patterns is possible. The following rough illustration of the construction of a reproducing kernel Hilbert space of functions aims at giving the reader an intuitive idea of the basic concept of the approach for establishing an optimization problem for classification and regression tasks. We mainly refer to the description given in [64]. A more common approach to establish the optimization problems that arise in the field of classification and regression tasks is by first introducing the ideas for linear classification and regression in a geometrical sense by means of separating hyperplanes and linear regression hyperplanes. After that, the so-called kernel trick allows for nonlinear classifiers and regression functions, in each case arriving at an optimization problem with linear inequality constraints. For more information on this approach see [46, 73, 42, 26].

Let $\mathcal{X}$ be a nonempty set. The function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be a kernel function if for all $x, y \in \mathcal{X}$ is satisfies

$$k(x, y) = \langle \phi(x), \phi(y) \rangle,$$

where $\phi : \mathcal{X} \to \mathcal{F}$ is a mapping from $\mathcal{X}$ to a so-called inner product feature space. We require k to satisfy the finitely positive semidefiniteness property. The function k satisfies the finitely positive semidefiniteness property if it is a symmetric function and the matrices $K = (k(x_i, x_j))_{i,j=1}^n \in \mathbb{R}^{n \times n}$ are positive semidefinite for all finite subsets $X = \{x_1, \ldots, x_n\} \subset \mathcal{X}$ and all $n \in \mathbb{N}$. The matrices K are called kernel matrices. It will be now demonstrated how this property characterizes kernel functions, i. e. if a function satisfies the finitely

positive semidefiniteness property it can be decomposed into a feature map ϕ into a Hilbert space of functions $\mathcal{F}$ applied to both its arguments followed by the evaluation of the inner product in $\mathcal{F}$ (cf. [64, Theorem 3.11]). Assume that $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is finitely positive semidefinite. We proceed by constructing a feature mapping ϕ into a Hilbert space for which k is the kernel. Consider the set of functions

$$\mathcal{F} = \left\{ \sum_{i=1}^{r} \alpha_i k(\cdot, x_i) : r \in \mathbb{N},\, x_i \in \mathcal{X},\, \alpha_i \in \mathbb{R},\, i = 1, \ldots, r \right\}. \tag{2.10}$$

where addition is defined by $(f + g)(x) = f(x) + g(x)$. Since we need an inner product in $\mathcal{F}$ we define for two functions $f, g \in \mathcal{F}$,

$$f(x) = \sum_{i=1}^{r} \alpha_i k(x_i, x), \quad g(x) = \sum_{i=1}^{l} \beta_j k(z_i, x),$$

the inner product to be

$$\langle f, g \rangle := \sum_{i=1}^{r} \sum_{j=1}^{l} \alpha_i \beta_j k(x_i, z_j). \tag{2.11}$$

The inner product defined by (2.11) fulfills all the properties required for an inner product (see [64]). Another property, namely the reproducing property of the kernel is also valid,

$$\langle k(\cdot, x), f \rangle = f(x).$$

The two additional properties of separability and completeness are also fulfilled (cf. [64]). Separability follows if the input space is countable or the kernel is continuous. Completeness is achieved, roughly speaking, if one adds all limit points of Cauchy sequences of functions to the set $\mathcal{F}$. We will denote the reproducing kernel Hilbert space associated with the kernel k by $\mathcal{H}_k$. The image of an input x under the mapping $\phi : \mathcal{X} \to \mathcal{H}_k$ is now specified by $\phi(x) = k(\cdot, x)$.

Next we derive an optimization problem that arises when a classification or regression task has to be solved based on a set of input patterns $X = \{x_1, \ldots, x_n\} \subset \mathcal{X}$, the corresponding observed values $\{y_1, \ldots, y_n\} \subset \mathbb{R}$ and a given kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ fulfilling the finitely positive semidefiniteness property. In the case of binary classification tasks the value y_i, $i = 1, \ldots, n$, denotes the class label of the corresponding input pattern x_i. In that case $y_i \in \{-1, 1\}$ for example. The aim is to find a function $f \in \mathcal{H}_k$ that appropriately approximates the given training data $\{(x_i, y_i) : i = 1, \ldots, n\} \subset \mathcal{X} \times \mathbb{R}$ and at the same time is smooth to guarantee that two similar inputs correspond to two similar outputs.

To be able to impose a penalty for predicting $f(x_i)$ while the true value is y_i, $i = 1, \ldots, n$, we introduce a loss function $v : \mathbb{R} \times \mathbb{R} \to \overline{\mathbb{R}}$ that is assumed to be proper and convex in its first variable. There exist many different loss functions to be taken into account, where we will make use of different loss functions in Chapter 3, each of them belonging to the class of loss functions for regression tasks or to the class of loss functions for the classification task. To guarantee smoothness of the resulting function f a smoothing functional $\Omega : \mathcal{H}_k \to \mathbb{R}$ is introduced which takes high values for non-smooth functions and low values for smooth ones.

The desired function f will be the optimal solution to the general Tikhonov regularization problem ([67])

$$\inf_{f \in \mathcal{H}_k} \left\{ C \sum_{i=1}^{n} v(f(x_i), y_i) + \frac{1}{2} \Omega(f) \right\}, \tag{2.12}$$

where $C > 0$ is the so-called regularization parameter controlling the trade-off between smoothness and accuracy of the resulting classifier or regression function, respectively. Under certain circumstances this would mean to solve an optimization problem in a possibly infinite dimensional space. However, the famous representer theorem (cf. [74]) allows a reformulation of this optimization problem in finite dimension. There, the smoothing functional is of the form $\Omega(f) = \frac{1}{2}\|f\|_{\mathcal{H}_k}^2$, where $\|\cdot\|_{\mathcal{H}_k}$ denotes the norm in the reproducing kernel Hilbert space $\mathcal{H}_k$. However, the starting point for our duality analysis in Chapter 3 will be a more general case by applying the generalized representer theorem stated in [62]. It says that, if $g : [0, \infty) \to \mathbb{R}$ is a strictly monotonically increasing function and we set $\Omega(f) = g(\|f\|_{\mathcal{H}_k})$, then for every minimizer f of the problem

$$\inf_{f \in \mathcal{H}_k} \left\{ C \sum_{i=1}^{n} v(f(x_i), y_i) + g\left(\|f\|_{\mathcal{H}_k}\right) \right\} \tag{2.13}$$

there exists a vector $c = (c_1, \ldots, c_n)^{\mathrm{T}} \in \mathbb{R}^n$ such that

$$f(\cdot) = \sum_{i=1}^{n} c_i k(\cdot, x_i). \tag{2.14}$$

The coefficients c_i are called expansion coefficients and all input points x_i for which $c_i \neq 0$ are the so-called support vectors. The existence of such a representation is essential to formulate an equivalent problem for (2.13) which will be the object of investigation via Fenchel-type duality. The norm $\|\cdot\|_{\mathcal{H}_k}$ induced by the inner product in the reproducing kernel Hilbert space $\mathcal{H}_k$ introduced by the inner product (2.11) becomes $\|f\|_{\mathcal{H}_k} = \sqrt{\langle f, f \rangle} = \sqrt{c^{\mathrm{T}} K c}$ for $f \in \mathcal{H}_k$. More than

this, from the finite representation we deduce that $f(x_i) = \sum_{j=1}^{n} c_j k(x_j, x_i) = (Kc)_i$ for all $x_i \in X$. Thus, we can formulate the equivalent optimization problem

$$(\mathrm{P_{SV}}) \qquad \inf_{c \in \mathbb{R}^n} \left\{ C \sum_{i=1}^{n} v((Kc)_i, y_i) + g\left(\sqrt{c^{\mathrm{T}} K c}\right) \right\}.$$

In Chapter 3 we will assign a Fenchel-type dual problem to a slightly modified primal problem and investigate duality statements and give optimality conditions. For several different loss functions the dual problems will be derived for both, classification and regression tasks by choosing the function g to be $g(\cdot) = \frac{1}{2}(\cdot)^2$ obtaining the standard form of the regularization term in $(\mathrm{P_{SV}})$. Finally, we numerically solve these dual problems for different tasks.

2.2.2 A general optimization problem

The general optimization problem we consider in this thesis is given by

$$(\mathrm{P_{gen}}) \qquad \inf_{x \in \mathbb{R}^n} \left\{ f(x) + \sum_{i=1}^{m} g_i(K_i x) \right\}.$$

Here, the function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is assumed to be proper, convex and lower semicontinuous. For $i = 1, \ldots, m$ the operators $K_i : \mathbb{R}^n \to \mathbb{R}^{k_i}$ are assumed to be linear operators. Finally, the functions $g_i : \mathbb{R}^{k_i} \to \overline{\mathbb{R}}$ are assumed to be proper, convex and lower semicontinuous functions. Thus, this problem is an unconstrained convex and nondifferentiable optimization problem for which first order methods involving a gradient step are not applicable. In this thesis we will establish an approach that utilizes Fenchel-type duality and a double smoothing technique to solve this problem at least approximately. This allows for applying a fast gradient method (cf. [50]) of rather simple structure and as we will see for a given accuracy $\varepsilon > 0$ we will obtain an approximate solution to $(\mathrm{P_{gen}})$ in $O(\frac{1}{\varepsilon} \ln\left(\frac{1}{\varepsilon}\right))$ iterations. For this approach the convergence of the dual objective value and of the norm of the gradient of the single smoothed dual objective can be shown. To obtain a primal solution, we can show that it is possible to construct one by using the sequences of points produced by the fast gradient scheme.

A special case arises when choosing $m = 1$ in $(\mathrm{P_{gen}})$. Then this problem becomes

$$\inf_{x \in \mathbb{R}^n} \{f(x) + g(Kx)\}, \tag{2.15}$$

where $K : \mathbb{R}^n \to \mathbb{R}^k$ is a linear operator, $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ and $g : \mathbb{R}^k \to \overline{\mathbb{R}}$ are proper, convex and lower semicontinuous functions. In [13] the authors develop

an algorithmic scheme for solving this problem from a more general point of view. The more general problem there is to find the zeros of sums of maximally monotone operators. This problem is addressed by applying Fenchel-type duality and solve it via a primal-dual splitting algorithm (cf. [3, 4, 55]). In this special case, since $\partial f : \mathbb{R}^n \to \mathbb{R}^n$ and $\partial g : \mathbb{R}^k \to \mathbb{R}^k$ are maximally monotone operators the general algorithmic scheme can be applied where it is possible to show convergence of the sequences of iterates to the optimal solution of the primal problem as well as to the optimal solution of the dual problem, respectively. Notice that in this case the algorithmic scheme in [28] is rediscovered. For the applications we will consider in this thesis it holds that $f \geq 0$ and $g \geq 0$, which are exactly the assumptions made in [28]. In our setting however we do not ask the functions in the objective to take only nonnegative values.

Having in mind the application of the double smoothing algorithm to image restoration tasks (cf. Section 5.1) we would like to mention here another algorithm that solves a minimization problem having the sum of two functions in the objective, namely the fast iterative shrinkage-thresholding algorithm (FISTA), an accelerated version of ISTA and variants of it (see [7, 11, 32]). In particular, in [7] they solve problems of the form

$$\inf_{x \in \mathbb{R}^n} \{f(x) + g(x)\}$$

where $g : \mathbb{R}^n \to \mathbb{R}$ is a convex and continuous function which is allowed to be non-smooth and $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function that is assumed to be continuously differentiable with Lipschitz continuous gradient. This algorithm can be applied to l_1-regularization problems having for example the squared norm function as second component. For this reason we can only compare the performance of the double smoothing algorithm and FISTA for this special choice of functions in Section 5.1. The main work to be done in each iteration there is the computation of the proximal point of the function g at a certain point (see [7] for details). Therefore, this algorithm belongs to the class of proximal algorithms. In the case $m = 1$ and f continuously differentiable and strongly convex in ($\mathrm{P}_{\mathrm{gen}}$) we also have to calculate the proximal point of g in each iteration of the double smoothing algorithm. Nevertheless, we will see that the double smoothing algorithm outperforms FISTA w. r. t. the image restoration task in Section 5.1. There we apply the double smoothing technique in its general version derived in Chapter 4, while it could be accelerated in this setting (see [22]) when having the sum of only two functions in the objective.

2.3 Fast Gradient Algorithm

In this section we will briefly introduce the gradient algorithm that we will use in Chapter 4 to solve the general optimization problem ($\mathrm{P}_{\mathrm{gen}}$) approximately via its

Fenchel-type dual problem. Therefore, consider the unconstrained optimization problem

$$\inf_{x\in\mathbb{R}^n}\{f(x)\}, \tag{2.16}$$

where the proper and convex function $f:\mathbb{R}^n\to\overline{\mathbb{R}}$ is assumed to be continuously differentiable on $\mathbb{R}^n$ with Lipschitz continuous gradient whose Lipschitz constant is $L>0$. Moreover, we assume f to be strongly convex with parameter $\gamma>0$ (see Definition 2.4). These assumptions on f ensure that the algorithmic scheme generates a sequence that converges to a global optimum, i. e. the first order optimality condition is sufficient for a global optimum.

The algorithmic scheme we are applying to solve (2.16) is introduced in [50]. We notice that we deal with a gradient method with constant step size $\frac{1}{L}$. This

Fast gradient scheme

Initialization: set $y^0=x^0\in\mathbb{R}^n$

Iteration $k\geq 0$: set

$$x^{k+1}=y^k-\frac{1}{L}\nabla f(y^k)$$

$$y^{k+1}=x^{k+1}+\frac{\sqrt{L}-\sqrt{\gamma}}{\sqrt{L}+\sqrt{\gamma}}\left(x^{k+1}-x^k\right)$$

step size is optimal for the gradient method for functions f that fulfill the above properties (see [50, Theorem 2.2.2]). For this algorithmic scheme we get the following estimate (cf. [50, Theorem 2.2.3]),

$$f(x^k)-f^*\leq\min\left\{\left(1-\sqrt{\frac{\gamma}{L}}\right)^k,\frac{4L}{(2\sqrt{L}+k\sqrt{\gamma})^2}\right\}\left(f(x^0)-f^*+\frac{\gamma}{2}\|x^0-x^*\|^2\right) \tag{2.17}$$

where $f^*=f(x^*)$ is the optimal objective value of f at the optimal solution $x^*\in\mathbb{R}^n$ of (2.16). Furthermore, for a function f being continuously differentiable and strongly convex with parameter $\gamma>0$ it holds for all $x\in\mathbb{R}^n$ (cf. [50, Theorem 2.1.8])

$$f(x)\geq f(x^*)+\frac{\gamma}{2}\|x-x^*\|^2, \tag{2.18}$$

where x^* is the unique minimizer for which it holds $\nabla f(x^*) = 0$. Therefore, for each $k \geq 0$ we have

$$\frac{\gamma}{2}\|x^k - x^*\|^2 \leq f(x^k) - f^*. \tag{2.19}$$

From (2.17) we get

$$f(x^k) - f^* \leq \Big(f(x^0) - f^* + \frac{\gamma}{2}\|x^0 - x^*\|^2\Big)\mathrm{e}^{-k\sqrt{\frac{\gamma}{L}}} \tag{2.20}$$

which can be further reformulated by using (2.19),

$$f(x^k) - f^* \leq 2\Big(f(x^0) - f^*\Big)\mathrm{e}^{-k\sqrt{\frac{\gamma}{L}}}. \tag{2.21}$$

Using (2.21) we get from (2.19) for f having Lipschitz continuous gradient with constant L,

$$\frac{\gamma}{2}\|x^k - x^*\|^2 \leq 2(f(x^0) - f^*)\mathrm{e}^{-k\sqrt{\frac{\gamma}{L}}}. \tag{2.22}$$

Since for a convex function f with Lipschitz continuous gradient with Lipschitz constant $L > 0$ it holds for all $x, y \in \mathbb{R}^n$ that

$$f(x) + \langle \nabla f(x), y - x\rangle + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2 \leq f(y)$$

(cf. [50, Theorem 2.1.5]) we obtain by setting $y = x^k$ and $x = x^*$ and by taking into account that $\nabla f(x^*) = 0$

$$\frac{1}{2L}\|\nabla f(x^k)\|^2 \leq f(x^k) - f^*. \tag{2.23}$$

Thus, we obtain an upper bound on the norm of the gradient applying (2.21) given by

$$\|\nabla f(x^k)\|^2 \leq 4L(f(x^0) - f^*)\mathrm{e}^{-k\sqrt{\frac{\gamma}{L}}}. \tag{2.24}$$

Chapter 3

Fenchel-type dual programs in machine learning

In this chapter we investigate the problem ($\mathrm{P_{SV}}$) introduced in Subsection 2.2.1. Other than in most literature we will not treat this problem based on Lagrange duality (see [46, 63, 69]). We rather employ Fenchel-type duality for such machine learning problems in analogy to the concept in [23] (see also [57]) but the more detailed formulation of the dual problem and the reduction of the dimension of the space of the dual variables compared to the one in [23] make it more suitable for calculations when dealing with concrete loss functions and for numerical implementations. By further reformulating the dual problems we actually solve numerically we obtain the standard form for these problems when applying Lagrange duality.

3.1 Duality and optimality conditions

In this section we will investigate the optimization problem ($\mathrm{P_{SV}}$) derived in section 2.2.1. The problem ($\mathrm{P_{SV}}$) will be slightly modified to match the general framework underlying the analysis concerning duality that is done in this section. We define the function $\tilde{g} : \mathbb{R} \to \overline{\mathbb{R}}$,

$$\tilde{g}(t) := \begin{cases} g(t), & \text{if } t \geq 0, \\ +\infty, & \text{otherwise,} \end{cases}$$

and introduce the problem

$$(\tilde{\mathrm{P}}_{\mathrm{SV}}) \qquad \inf_{c \in \mathbb{R}^n} \left\{ C \sum_{i=1}^{n} v((Kc)_i, y_i) + \tilde{g}\left(\sqrt{c^{\mathrm{T}} K c}\right) \right\}.$$

By assumption the function $v : \mathbb{R} \times \mathbb{R} \to \overline{\mathbb{R}}$ is proper and convex in its first variable. Remember that the function $g : [0, +\infty) \to \mathbb{R}$ is assumed to be

strictly monotonically increasing (cf. 2.2.1). Therefore, the problem $\left(\tilde{\mathrm{P}}_{\mathrm{SV}}\right)$ is a convex optimization problem which is, depending on the choice of the particular loss function v, not necessarily differentiable. To this problem we assign the Fenchel-type dual problem

$$\left(\tilde{\mathrm{D}}_{\mathrm{SV}}\right) \qquad \sup_{\substack{P\in\mathbb{R}^n,\\ P=(P_1,\ldots,P_n)^{\mathrm{T}}}} \left\{-C\sum_{i=1}^{n}\left(v(\cdot,y_i)\right)^*\left(-\frac{P_i}{C}\right) - \tilde{g}^*(\sqrt{P^{\mathrm{T}}KP})\right\}. \tag{3.1}$$

In the following we will prove that weak duality always holds between the primal-dual pair $\left(\tilde{\mathrm{P}}_{\mathrm{SV}}\right) - \left(\tilde{\mathrm{D}}_{\mathrm{SV}}\right)$ and, if a certain qualification condition is satisfied, even strong duality can be assured. The next lemma establishes a certain factorization of a real symmetric and positive semidefinite matrix $M \in \mathbb{R}^{n\times n}$ (cf. [41, Theorem 7.2.6]) which will be used in the proofs of the following two theorems.

Lemma 3.1. *Let $M \in \mathbb{R}^{n\times n}$ be a real symmetric positive semidefinite matrix and let $k \geq 1$ be a given integer. Then there exists a unique positive semidefinite and symmetric matrix $B \in \mathbb{R}^{n\times n}$ such that $B^k = M$.*

By denoting by $v\left(\tilde{\mathrm{P}}_{\mathrm{SV}}\right)$ and $v\left(\tilde{\mathrm{D}}_{\mathrm{SV}}\right)$ the optimal objective values of $\left(\tilde{\mathrm{P}}_{\mathrm{SV}}\right)$ and $\left(\tilde{\mathrm{D}}_{\mathrm{SV}}\right)$, respectively, the following theorem states that weak duality always holds between $\left(\tilde{\mathrm{P}}_{\mathrm{SV}}\right)$ and $\left(\tilde{\mathrm{D}}_{\mathrm{SV}}\right)$.

Theorem 3.2. *For $\left(\tilde{P}_{SV}\right)$ and $\left(\tilde{D}_{SV}\right)$ weak duality holds, i. e. $v\left(\tilde{P}_{SV}\right) \geq v\left(\tilde{D}_{SV}\right)$.*

Proof. Let $c \in \mathbb{R}^n$ and $P = (P_1, \ldots, P_n)^{\mathrm{T}} \in \mathbb{R}^n$. From the Young-Fenchel inequality we get for all $i = 1, \ldots, n$,

$$v((Kc)_i, y_i) + \left(v(\cdot, y_i)\right)^*\left(-\frac{P_i}{C}\right) - (Kc)_i\left(-\frac{P_i}{C}\right) \geq 0$$

and therefore, by summing up the inequalities for all $i = 1, \ldots, n$ and multiplying by $C > 0$,

$$C\sum_{i=1}^{n} v((Kc)_i, y_i) + C\sum_{i=1}^{n}\left(v(\cdot, y_i)\right)^*\left(-\frac{P_i}{C}\right) + \sum_{i=1}^{n}(Kc)_i P_i \geq 0. \tag{3.2}$$

Again by the Young-Fenchel inequality we have for the function $\tilde{g}$ and its conjugate function $\tilde{g}^*$

$$\tilde{g}\left(\sqrt{c^{\mathrm{T}}Kc}\right) + \tilde{g}^*\left(\sqrt{P^{\mathrm{T}}KP}\right) - \sqrt{c^{\mathrm{T}}Kc}\sqrt{P^{\mathrm{T}}KP} \geq 0. \tag{3.3}$$

Summing up (3.2) and (3.3) we get

$$0 \leq C\sum_{i=1}^{n} v((Kc)_i, y_i) + C\sum_{i=1}^{n}\left(v(\cdot, y_i)\right)^*\left(-\frac{P_i}{C}\right) + \sum_{i=1}^{n}(Kc)_i P_i$$

$$+ \tilde{g}\left(\sqrt{c^{\mathrm{T}}Kc}\right) + \tilde{g}^*\left(\sqrt{P^{\mathrm{T}}KP}\right) - \sqrt{c^{\mathrm{T}}Kc}\sqrt{P^{\mathrm{T}}KP}$$

or

$$C\sum_{i=1}^{n} v((Kc)_i, y_i) + \tilde{g}\left(\sqrt{c^{\mathrm{T}}Kc}\right) \geq -C\sum_{i=1}^{n}\left(v(\cdot, y_i)\right)^*\left(-\frac{P_i}{C}\right) - \tilde{g}^*\left(\sqrt{P^{\mathrm{T}}KP}\right) - \left(\sum_{i=1}^{n}(Kc)_i P_i - \sqrt{c^{\mathrm{T}}Kc}\sqrt{P^{\mathrm{T}}KP}\right) \quad (3.4)$$

From Lemma 3.1 we have that there exists a real symmetric and positive semidefinite matrix $L \in \mathbb{R}^{n\times n}$ such that $K = LL$. Furthermore, by applying the Cauchy-Schwarz inequality, we get

$$\begin{aligned} P^{\mathrm{T}}Kc - \sqrt{c^{\mathrm{T}}Kc}\sqrt{P^{\mathrm{T}}KP} &= (LP)^{\mathrm{T}}(Lc) - \sqrt{(Lc)^{\mathrm{T}}(Lc)}\sqrt{(LP)^{\mathrm{T}}(LP)} \\ &= \langle LP, Lc\rangle - \|Lc\|\|LP\| \leq 0. \end{aligned}$$

Thus, incorporating this observation in (3.4) we finally get

$$C\sum_{i=1}^{n} v((Kc)_i, y_i) + \tilde{g}\left(\sqrt{c^{\mathrm{T}}Kc}\right) \geq -C\sum_{i=1}^{n}\left(v(\cdot, y_i)\right)^*\left(-\frac{P_i}{C}\right) - \tilde{g}^*\left(\sqrt{P^{\mathrm{T}}KP}\right)$$

which means that $v\big(\tilde{\mathrm{P}}_{\mathrm{SV}}\big) \geq v\big(\tilde{\mathrm{D}}_{\mathrm{SV}}\big)$. □

In order to ensure strong duality between the primal-dual pair $\big(\tilde{\mathrm{P}}_{\mathrm{SV}}\big)$ and $\big(\tilde{\mathrm{D}}_{\mathrm{SV}}\big)$ we impose the qualification condition

$$(QC) \qquad \mathrm{Im}K \cap \prod_{i=1}^{n} \mathrm{ri}(\mathrm{dom}\, v(\cdot, y_i)) \neq \emptyset. \quad (3.5)$$

Theorem 3.3. *If (QC) is fulfilled, then it holds $v\big(\tilde{P}_{SV}\big) = v\big(\tilde{D}_{SV}\big)$ and $\big(\tilde{D}_{SV}\big)$ has an optimal solution.*

Proof. First, we define $h : \mathbb{R}^n \to \overline{\mathbb{R}}$, $h(x) = (\tilde{g} \circ \beta)(x)$, where $\beta : \mathbb{R}^n \to \mathbb{R}$, $\beta(x) = \sqrt{x^{\mathrm{T}}Kx}$. Furthermore, we define $v_i : \mathbb{R}^n \to \overline{\mathbb{R}}$, $v_i(x) := v(x_i, y_i)$, then we have

$$v\big(\tilde{\mathrm{P}}_{\mathrm{SV}}\big) = \inf_{c\in\mathbb{R}^n}\left\{\left(\sum_{i=1}^{n} Cv_i\right)(Kc) + h(c)\right\}.$$

Since K is assumed to be positive semidefinite, $\beta(x) \geq 0$ for all $x \in \mathbb{R}^n$ and therefore, $\mathrm{dom}\, h = \mathbb{R}^n$ and $K(\mathrm{dom}\, h) = K(\mathbb{R}^n) = \mathrm{Im}K$. By taking into consideration that $\mathrm{dom}\,\left(\sum_{i=1}^{n} Cv_i\right) = \prod_{i=1}^{n} \mathrm{dom}\,\left(v(\cdot, y_i)\right)$ we have that

$$K(\mathrm{ri}(\mathrm{dom}\, h)) \cap \mathrm{ri}\left(\mathrm{dom}\,\left(\sum_{i=1}^{n} Cv_i\right)\right) = \mathrm{Im}K \cap \prod_{i=1}^{n} \mathrm{ri}\left(\mathrm{dom}\, v(\cdot, y_i)\right) \neq \emptyset.$$

This means that $v\big(\tilde{\mathrm{P}}_{\mathrm{SV}}\big) < +\infty$. Taking into account Lemma 3.1 we have that $\beta(x) = \sqrt{x^{\mathrm{T}}Kx} = \sqrt{(Lx)^{\mathrm{T}}(Lx)} = \|Lx\|$ is a convex function and so is h. Then we have that (see [12, Theorem 2.1])

$$v\big(\tilde{\mathrm{P}}_{\mathrm{SV}}\big) = \sup_{P\in\mathbb{R}^n} \left\{ -\left(\sum_{i=1}^{n} Cv_i\right)^* (-P) - h^*(KP) \right\}.$$

Next we calculate the conjugate function of h. For all $z \in \mathbb{R}^n$ we have from [16] that

$$\begin{aligned} h^*(z) = (\tilde{g}\circ\beta)^*(z) &= \min_{q\geq 0} \{\tilde{g}^*(q) + (q\beta)^*(z)\} \\ &= \min\left\{ \tilde{g}^*(0) + \delta_{\{0\}}(z), \inf_{q>0} \left\{ \tilde{g}^*(q) + q\beta^*\left(\frac{1}{q}z\right)\right\}\right\}. \quad (3.6) \end{aligned}$$

Furthermore,

$$\beta^*(z) = \begin{cases} 0, & \text{if } \sqrt{z^{\mathrm{T}}K^-z} \leq 1 \text{ and } z \in \mathrm{Im}K, \\ +\infty, & \text{else.} \end{cases}$$

where K^- denotes the Moore-Penrose pseudo inverse (cf. [8]) of K. To see this, consider the following three cases.

(i) Let $z \in \mathrm{Im}K$ and $\sqrt{z^{\mathrm{T}}K^-z} \leq 1$. Then $\exists a \in \mathbb{R}^n$ such that $z = Ka$ and $\sqrt{(Ka)^{\mathrm{T}}K^-(Ka)} = \sqrt{a^{\mathrm{T}}KK^-Ka} = \sqrt{a^{\mathrm{T}}Ka} \leq 1$. By applying Lemma 3.1 and the Cauchy-Schwarz inequality, we get

$$\begin{aligned} \beta^*(z) &= \sup_{c\in\mathbb{R}^n} \left\{ a^{\mathrm{T}}Kc - \sqrt{c^{\mathrm{T}}Kc}\right\} \leq \sup_{c\in\mathbb{R}^n} \left\{ a^{\mathrm{T}}Kc - \sqrt{c^{\mathrm{T}}Kc}\sqrt{a^{\mathrm{T}}Ka}\right\} \\ &= \sup_{c\in\mathbb{R}^n} \left\{ (La)^{\mathrm{T}}(Lc) - \|Lc\|\|La\|\right\} \leq 0. \end{aligned}$$

The supremum is attained for $c = 0$ and is equal to 0, i. e. $\beta^*(z) = 0$.

(ii) Let $z \notin \mathrm{Im}K$. Then, z can be represented as $z = u + Ka$, where $u \in \mathrm{Ker}K$, $u \neq 0$, and $a \in \mathbb{R}^n$ and

$$\begin{aligned} \beta^*(z) &= \sup_{c\in\mathbb{R}^n} \left\{ z^{\mathrm{T}}c - \sqrt{c^{\mathrm{T}}Kc}\right\} = \sup_{c\in\mathbb{R}^n} \left\{ u^{\mathrm{T}}c + a^{\mathrm{T}}Kc - \sqrt{c^{\mathrm{T}}Kc}\right\} \\ &\geq \sup_{t>0} \left\{ t\|u\|^2 + ta^{\mathrm{T}}Ku - t\sqrt{u^{\mathrm{T}}Ku}\right\} = \sup_{t>0} \left\{ t\|u\|^2\right\} = +\infty, \end{aligned}$$

since $\|u\|^2 > 0$, $Ku = 0$ and by setting $c := tu$, $t > 0$.

(iii) Let $z \in \mathrm{Im}K$ and $\sqrt{z^{\mathrm{T}}K^-z} > 1$, i. e. $\exists a \in \mathbb{R}^n$ such that $z = Ka$ and $\sqrt{z^{\mathrm{T}}K^-z} = \sqrt{a^{\mathrm{T}}Ka} > 1$. Then,

$$\beta^*(z) = \sup_{c\in\mathbb{R}^n} \left\{ a^{\mathrm{T}}Kc - \sqrt{c^{\mathrm{T}}Kc}\right\} \geq \sup_{t>0} \left\{ ta^{\mathrm{T}}Ka - t\sqrt{a^{\mathrm{T}}Ka}\right\}$$

$$= \sup_{t>0} \left\{ t\sqrt{a^{\mathrm{T}}Ka} \left(\sqrt{a^{\mathrm{T}}Ka} - 1 \right) \right\} = +\infty$$

and we conclude that

$$\beta^* \begin{pmatrix} 1 \\ \overline{q} z \end{pmatrix} = \begin{cases} 0, & \text{if } \sqrt{z^{\mathrm{T}}K^- z} \leq q \text{ and } z \in \mathrm{Im}K, \\ +\infty, & \text{else.} \end{cases}$$

and

$$h^*(z) = \min \left\{ \tilde{g}^*(0) + \delta_{\{0\}}(z), \inf_{q>0} \left\{ \tilde{g}^*(q) + q\beta^* \left(\frac{1}{q} z \right) \right\} \right\} = \inf_{\substack{q\geq 0, \\ q \geq \sqrt{z^{\mathrm{T}}K^- z}, \\ z \in \mathrm{Im}K}} \{\tilde{g}^*(q)\}$$

$$= - \sup_{\substack{q\geq 0, \\ q \geq \sqrt{z^{\mathrm{T}}K^- z}, \\ z \in \mathrm{Im}K}} \{-\tilde{g}^*(q)\}.$$

Now we have

$$-h^*(KP) = \sup_{\substack{q\geq 0, \\ q \geq \sqrt{P^{\mathrm{T}}KP}}} \{-\tilde{g}^*(q)\}.$$

which yields

$$\begin{aligned} v\big(\tilde{\mathrm{P}}_{\mathrm{SV}}\big) &= \sup_{P\in\mathbb{R}^n} \left\{ - \left(\sum_{i=1}^n Cv_i \right)^* (-P) - h^*(KP) \right\} \\ &= \sup_{P\in\mathbb{R}^n} \left\{ - \left(\sum_{i=1}^n Cv_i \right)^* (-P) + \sup_{\substack{q\geq 0, \\ q \geq \sqrt{P^{\mathrm{T}}KP}}} \{-\tilde{g}^*(q)\} \right\} \\ &= \sup_{\substack{P\in\mathbb{R}^n \\ q\geq 0, q \geq \sqrt{P^{\mathrm{T}}KP}}} \left\{ - \left(\sum_{i=1}^n Cv_i \right)^* (-P) - \tilde{g}^*(q) \right\}. \end{aligned} \tag{3.7}$$

To reformulate the last expression we show that $\tilde{g}^*$ is monotonically increasing. For all $t_1, t_2 \in \mathbb{R}$ such that $0 \leq t_1 \leq t_2$ we observe $\tilde{g}^*(t_1) = \sup_{a\geq 0}\{at_1 - g(a)\} \leq \sup_{a\geq 0}\{at_2 - g(a)\} = \tilde{g}^*(t_2)$. Now (3.7) becomes

$$v\big(\tilde{\mathrm{P}}_{\mathrm{SV}}\big) = \sup_{P\in\mathbb{R}^n} \left\{ - \left(\sum_{i=1}^n Cv_i \right)^* (-P) - \tilde{g}^*(\sqrt{P^{\mathrm{T}}KP}) \right\}$$

and that there exists a $\bar{P} \in \mathbb{R}^n$ (see [12, Theorem 2.1]) such that

$$v\big(\tilde{\mathrm{P}}_{\mathrm{SV}}\big) = - \left(\sum_{i=1}^n Cv_i \right)^* (-\bar{P}) - \tilde{g}^*(\sqrt{\bar{P}^{\mathrm{T}}K\bar{P}})$$

$$= -C\left(\sum_{i=1}^{n} v_i\right)^*\left(-\frac{1}{C}\bar{P}\right) - \tilde{g}^*(\sqrt{\bar{P}^{\mathrm{T}}K\bar{P}}).$$

As from (QC) we have that $\cap_{i=1}^{n}\mathrm{ri}(\mathrm{dom}\, v_i) = \prod_{i=1}^{n}\mathrm{ri}(\mathrm{dom}\, v(\cdot, y_i)) \neq \emptyset$ it follows (cf. [58, Theorem 16.4]) that there exist $\bar{P}^i \in \mathbb{R}^n$, $i = 1, \ldots, n$, with $\sum_{i=1}^{n}\bar{P}^i = \bar{P}$, such that

$$\left(\sum_{i=1}^{n} v_i\right)^*\left(-\frac{1}{C}\bar{P}\right) = \sum_{i=1}^{n} v_i^*\left(-\frac{1}{C}\bar{P}^i\right)$$

and, therefore,

$$v\big(\tilde{\mathrm{P}}_{\mathrm{SV}}\big) = -C\sum_{i=1}^{n} v_i^*\left(-\frac{1}{C}\bar{P}^i\right) - \tilde{g}^*\left(\sqrt{\left(\sum_{i=1}^{n}\bar{P}^i\right)^{\mathrm{T}} K\left(\sum_{i=1}^{n}\bar{P}^i\right)}\right).$$

Further, for all $i = 1, \ldots, n$ it holds

$$\begin{aligned} v_i^*\left(-\frac{1}{C}\bar{P}^i\right) &= \sup_{z\in\mathbb{R}^n}\left\{-\frac{1}{C}\left(\bar{P}^i\right)^{\mathrm{T}} z - v(z_i, y_i)\right\} \\ &= \begin{cases} (v(\cdot, y_i))^*\left(-\frac{1}{C}\bar{P}_i^i\right), & \text{if } \bar{P}_j^i = 0, \forall j \neq i, \\ +\infty, & \text{else.} \end{cases} \end{aligned}$$

Since the optimal objective value of $\big(\tilde{\mathrm{P}}_{\mathrm{SV}}\big)$ is finite, by defining $\bar{P}_i := \bar{P}_i^i$ for $i = 1, ..., n$, one has $\sum_{i=1}^{n}\bar{P}^i = (\bar{P}_1, \ldots, \bar{P}_n)^T \in \mathbb{R}^n$ and

$$v\big(\tilde{\mathrm{P}}_{\mathrm{SV}}\big) = -C\sum_{i=1}^{n}(v(\cdot, y_i))^*\left(-\frac{1}{C}\bar{P}_i\right) - \tilde{g}^*\left(\sqrt{\bar{P}^{\mathrm{T}}K\bar{P}}\right),$$

where $\bar{P} := (\bar{P}_1, \ldots, \bar{P}_n)^T$. This, along with Theorem 3.2, provides the desired result, $\bar{P}$ being an optimal solution to $\big(\tilde{\mathrm{D}}_{\mathrm{SV}}\big)$. □

The next theorem furnishes the necessary and sufficient optimality conditions for the primal-dual pair $\big(\tilde{\mathrm{P}}_{\mathrm{SV}}\big)$-$\big(\tilde{\mathrm{D}}_{\mathrm{SV}}\big)$.

Theorem 3.4. *Let (QC) be fulfilled. Then $\bar{c} \in \mathbb{R}^n$ is an optimal solution for $\big(\tilde{P}_{SV}\big)$ if and only if there exists an optimal solution $\bar{P} \in \mathbb{R}^n$ to $\big(\tilde{D}_{SV}\big)$ such that*

(i) $-\frac{\bar{P}_i}{C} \in \partial v(\cdot, y_i)\,((K\bar{c})_i)$, $i = 1, \ldots, n$,

(ii) $\tilde{g}(\sqrt{\bar{c}^{\mathrm{T}}K\bar{c}}) + \tilde{g}^*(\sqrt{\bar{P}^{\mathrm{T}}K\bar{P}}) = \bar{P}^{\mathrm{T}}K\bar{c}$.

Proof. From Theorem 3.3 we get the existence of an optimal solution $\bar{P} \in \mathbb{R}^n$ to $\left(\tilde{\mathrm{D}}_{\mathrm{SV}}\right)$ such that

$$C\left[\sum_{i=1}^{n} v((K\bar{c})_i, y_i) + \sum_{i=1}^{n} \left(v(\cdot, y_i)\right)^* \left(-\frac{1}{C}\bar{P}_i\right) - \sum_{i=1}^{n} (K\bar{c})_i \left(-\frac{1}{C}\bar{P}_i\right)\right] + \tilde{g}\left(\sqrt{\bar{c}^{\mathrm{T}} K \bar{c}}\right) + \tilde{g}^*\left(\sqrt{\bar{P}^{\mathrm{T}} K \bar{P}}\right) - \bar{P}^{\mathrm{T}} K \bar{c} = 0.$$

By the Young-Fenchel inequality (2.8) we get that the expression in square brackets as well as the three summands on the second line of the above formula are greater or equal to zero. Thus, both of them are zero to fulfill the above equality which is equivalent to

$$\begin{cases} v((K\bar{c})_i, y_i) + \left(v(\cdot, y_i)\right)^* \left(-\frac{1}{C}\bar{P}_i\right) = (K\bar{c})_i \left(-\frac{1}{C}\bar{P}_i\right), \quad i = 1, \ldots, n, \\ \tilde{g}\left(\sqrt{\bar{c}^{\mathrm{T}} K \bar{c}}\right) + \tilde{g}^*\left(\sqrt{\bar{P}^{\mathrm{T}} K \bar{P}}\right) = \bar{P}^{\mathrm{T}} K \bar{c}, \end{cases} \tag{3.8a}$$

and we get the optimality conditions (i) and (ii) by regarding the characterization of elements in the subdifferential given by (2.9). □

In the following sections we will derive several optimization problems as special instances of $\left(\tilde{\mathrm{P}}_{\mathrm{SV}}\right)$ for different choices of loss functions v, each of them designed for the classification or the regression task, respectively. Both cases, however, have in common the choice of the regularization term in $\left(\tilde{\mathrm{P}}_{\mathrm{SV}}\right)$, i. e. the concrete form of the function $\tilde{g} : \mathbb{R} \to \overline{\mathbb{R}}$ which is specified to be

$$\tilde{g}(t) = \begin{cases} \frac{1}{2}t^2, & \text{if } t \geq 0, \\ +\infty, & \text{else,} \end{cases} \tag{3.9}$$

and corresponds of considering the squared norm of $f \in \mathcal{H}_k$ as regularization functional, i. e. $\Omega(f) = \frac{1}{2}\|f\|_{\mathcal{H}_k}^2$ in (2.12), while the particular instance (3.9) of $\tilde{g}$ in $\left(\tilde{\mathrm{P}}_{\mathrm{SV}}\right)$ corresponds of choosing $g(t) = \frac{1}{2}t^2$ in $(\mathrm{P}_{\mathrm{SV}})$, a commonly used regularization term in literature. The primal optimization problem associated with this choice of $\tilde{g}$ is now given by

$$\left(\bar{\mathrm{P}}_{\mathrm{SV}}\right) \qquad \inf_{c \in \mathbb{R}^n} \left\{ C \sum_{i=1}^{n} v((Kc)_i, y_i) + \frac{1}{2} c^{\mathrm{T}} K c \right\}, \tag{3.10}$$

since $\sqrt{c^{\mathrm{T}} K c} \geq 0$ for all $c \in \mathbb{R}^n$. The corresponding dual problem is obtained by considering the conjugate function $\tilde{g}^*$ of $\tilde{g}$, which is

$$\tilde{g}^*(y) = \sup_{x \in \operatorname{dom} \tilde{g}} \{xy - \tilde{g}(x)\} = \begin{cases} \frac{1}{2}y^2, & \text{if } y \geq 0, \\ 0, & \text{if } y < 0, \end{cases}$$

then we have

$$\left(\bar{\mathrm{D}}_{\mathrm{SV}}\right) \qquad \sup_{\substack{P\in\mathbb{R}^n,\\ P=(P_1,\ldots,P_n)^{\mathrm{T}}}} \left\{-C\sum_{i=1}^{n}\left(v(\cdot,y_i)\right)^*\left(-\frac{P_i}{C}\right)-\frac{1}{2}P^{\mathrm{T}}KP\right\}, \tag{3.11}$$

since $\sqrt{x^{\mathrm{T}}Kx}\geq 0$ for all $x\in\mathbb{R}^n$ and for K being a kernel matrix. For this particular instance of the general primal-dual pair $\left(\tilde{\mathrm{P}}_{\mathrm{SV}}\right)$-$\left(\tilde{\mathrm{D}}_{\mathrm{SV}}\right)$ clearly strong duality holds as a consequence of Theorem 3.3 when imposing the corresponding regularity condition. The corresponding adapted optimality conditions are stated in the next corollary.

Corollary 3.5. *Let (QC) be fulfilled. Then $\bar{c}\in\mathbb{R}^n$ is an optimal solution for the problem $\left(\bar{P}_{SV}\right)$ if and only if there exists an optimal solution $\bar{P}\in\mathbb{R}^n$ to the problem $\left(\bar{D}_{SV}\right)$ such that*

(i) $-\frac{\bar{P}_i}{C}\in\partial v(\cdot,y_i)((K\bar{c})_i)$, $i=1,\ldots,n$,

(ii) $K(\bar{c}-\bar{P})=0$.

Proof. The statement (i) is the same as in Theorem 3.4. We show (ii). Considering (3.8a) and taking into account the particular choice of $\tilde{g}$ (cf. (3.9)), we have

$$\frac{1}{2}\bar{c}^{\mathrm{T}}K\bar{c}+\frac{1}{2}\bar{P}^{\mathrm{T}}K\bar{P}-\bar{P}^{\mathrm{T}}K\bar{c}=0$$

which is, since K is a kernel matrix, equivalent to

$$\frac{1}{2}(\bar{c}-\bar{P})^{\mathrm{T}}K(\bar{c}-\bar{P})=0. \tag{3.12}$$

Thus, $\bar{c}-\bar{P}$ is a global minimum of the convex function $p\mapsto\frac{1}{2}p^{\mathrm{T}}Kp$, which means that (3.12) is nothing else than $K(\bar{c}-\bar{P})=0$. □

The case where the kernel matrix $K\in\mathbb{R}^{n\times n}$ is positive definite allows further useful reformulations of the optimality conditions.

Remark 3.6. If K is positive definite, then, due to the fact that $v(\cdot,y_i)$ is proper and convex for all $i=1,\ldots,n$, the qualification condition (QC) is automatically fulfilled. Thus, according to Theorem 3.5, $\bar{c}\in\mathbb{R}^n$ is an optimal solution for problem $\left(\bar{\mathrm{P}}_{\mathrm{SV}}\right)$ if and only if there exists an optimal solution $\bar{P}\in\mathbb{R}^n$ to $\left(\bar{\mathrm{D}}_{\mathrm{SV}}\right)$ such that

(i) $-\frac{\bar{P}_i}{C}\in\partial v(\cdot,y_i)((K\bar{c})_i)$, $i=1,\ldots,n$,

(ii) $\bar{c}=\bar{P}$.

Remark 3.7. If K is positive definite, then the function $c \mapsto \frac{1}{2}c^{\mathrm{T}}Kc$ is strongly convex (on $\mathbb{R}^n$). Consequently, if $v(\cdot, y_i)$, $i = 1, \ldots, n$, is, additionally, lower semicontinuous, the optimization problem $\left(\bar{\mathrm{P}}_{\mathrm{SV}}\right)$ has a unique optimal solution (see, for instance, [38, Satz 6.33]). Further, due to the fact that $P \mapsto \frac{1}{2}P^T KP$ is strictly convex (on $\mathbb{R}^n$), one can see that the dual problem $\left(\bar{\mathrm{D}}_{\mathrm{SV}}\right)$ has at most one optimal solution. This yields, due to Remark 3.6, that whenever K is positive definite and $v(\cdot, y_i)$ is lower semicontinuous, for $i = 1, ..., n$, in order to solve $\left(\bar{\mathrm{P}}_{\mathrm{SV}}\right)$ one can equivalently solve $\left(\bar{\mathrm{D}}_{\mathrm{SV}}\right)$ which in this case has an unique optimal solution $\bar{P}$, this being also the unique optimal solution of $\left(\bar{\mathrm{P}}_{\mathrm{SV}}\right)$.

In the following sections we will consider the particular instances of primal and dual problems $\left(\bar{\mathrm{P}}_{\mathrm{SV}}\right)$ and $\left(\bar{\mathrm{D}}_{\mathrm{SV}}\right)$, respectively, arising with the choice of different loss functions. The particular choice of the regularization term is due to the more easy numerical solvability of the resulting dual optimization problems.

3.2 Dual programs for the classification task

In this section we deal with particular instances of the general model described in Section 3.1 and construct, for three particular loss functions, the corresponding dual problems. We apply these three dual problems in order to solve a classification problem on a data set of images, as we will show in Section 3.3.

3.2.1 Hinge loss

The first loss function we consider here is the hinge loss $v_{\mathrm{hl}} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, defined as

$$v_{\mathrm{hl}}(a, y) = (1 - ay)_+ = \max\{0, 1 - ay\}, \tag{3.13}$$

which is a proper, convex and lower semicontinuous function in its first component, while (QC) is obviously fulfilled. The primal optimization problem $\left(\bar{\mathrm{P}}_{\mathrm{SV}}\right)$ becomes in this case

$$(\mathrm{P}_{\mathrm{hl}}) \qquad \inf_{c \in \mathbb{R}^n} \left\{ C \sum_{i=1}^{n} \left(1 - (Kc)_i y_i\right)_+ + \frac{1}{2} c^{\mathrm{T}} Kc \right\}.$$

To obtain the dual problem $(\mathrm{D}_{\mathrm{hl}})$ of $(\mathrm{P}_{\mathrm{hl}})$ for this special loss function, we use the Lagrange technique in order to calculate the conjugate function of $v_{\mathrm{hl}}(\cdot, y_i)$, for $i = 1, ..., n$. For $z \in \mathbb{R}$ and $i = 1, ..., n$ we have

$$\begin{aligned} -\left(v_{\mathrm{hl}}(\cdot, y_i)\right)^*(z) &= -\sup_{a \in \mathbb{R}} \{za - (1 - ay_i)_+\} = \inf_{\substack{a,t \in \mathbb{R}, \\ t \geq 0,\, t \geq 1 - ay_i}} \{-za + t\} \\ &= \sup_{k \geq 0,\, r \geq 0} \left\{ \inf_{a,t \in \mathbb{R}} \{ -za + t + k(1 - ay_i - t) - rt \} \right\} \end{aligned}$$

$$= \sup_{k\geq 0,\, r\geq 0} \left\{ \inf_{a\in\mathbb{R}} \{-za - kay_i\} + \inf_{t\in\mathbb{R}} \{t - kt - rt\} + k \right\}$$

$$= \sup_{\substack{k\geq 0,\, r\geq 0,\\ k+r=1,\\ z+ky_i=0}} k = \sup_{\substack{k\in[0,1],\\ k=-zy_i}} k = \begin{cases} -zy_i, & \text{if } zy_i \in [-1,0], \\ -\infty, & \text{otherwise.} \end{cases}$$

Note that in the calculations above we used the fact that the labels y_i, $i = 1, \ldots, n$, can only take the values $+1$ or -1 for the binary classification task we will consider in Section 3.3. With the above formula we obtain the following dual problem

$$(\mathrm{D_{hl}}) \qquad \sup_{\substack{P\in\mathbb{R}^n,\\ P_iy_i\in[0,C],\, i=1,\ldots,n}} \left\{ \sum_{i=1}^{n} P_iy_i - \frac{1}{2}P^{\mathrm{T}}KP \right\}$$

or, by reformulating it as an infimum problem and keeping notation,

$$(\mathrm{D_{hl}}) \qquad \inf_{\substack{P\in\mathbb{R}^n,\\ P_iy_i\in[0,C],\, i=1,\ldots,n}} \left\{ \frac{1}{2}P^{\mathrm{T}}KP - \sum_{i=1}^{n} P_iy_i \right\}.$$

By defining the vector $\alpha = (\alpha_1, \ldots, \alpha_n)^{\mathrm{T}} \in \mathbb{R}^n$, $\alpha_i := P_iy_i$, $i = 1, \ldots, n$, the dual problem can equivalently be written as

$$(\mathrm{D_{hl}}) \qquad \inf_{\alpha_i\in[0,C],\, i=1,\ldots,n} \left\{ \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_jy_iy_jK_{ij} - \sum_{i=1}^{n} \alpha_i \right\},$$

a representation which is recognized to be the commonly used form of the dual problem to $(\mathrm{P_{hl}})$ in the literature (see for example [63, 46]).

3.2.2 Generalized hinge loss

Beside the hinge loss, the binary image classification task has been performed for two other loss functions, as we point out in Section 3.3. They both represent particular instances of the generalized hinge loss $v^u_{ghl} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$,

$$v^u_{\mathrm{ghl}}(a, y) = (1 - ay)^u_+, \tag{3.14}$$

where $u > 1$. The generalized hinge loss function is proper, convex and lower semicontinuous in its first component, too, while the qualification condition (QC) is again obviously fulfilled. The primal problem to which this loss function gives rise to reads

$$(\mathrm{P^u_{ghl}}) \qquad \inf_{c\in\mathbb{R}^n} \left\{ C\sum_{i=1}^{n} (1 - (Kc)_iy_i)^u_+ + \frac{1}{2}c^{\mathrm{T}}Kc \right\}.$$

To obtain its dual problem we need the conjugate function of $v^u_{\mathrm{ghl}}(\cdot, y_i)$ for $i = 1, ..., n$. For all $z \in \mathbb{R}$ and all $i = 1, ..., n$ we have

$$-\big(v^u_{\mathrm{ghl}}(\cdot, y_i)\big)^*(z) = -\sup_{a\in\mathbb{R}} \left\{za - (1 - ay_i)^u_+\right\} = \inf_{\substack{a,t\in\mathbb{R},\\ t\geq 1-ay_i}} \left\{-za + t^u + \delta_{[0,+\infty)}(t)\right\}.$$

By taking into account that the function $t \mapsto t^u + \delta_{[0,+\infty)}(t)$ is convex, we can make again use of Lagrange duality, which provides the following formula for the conjugate of $v^u_{\mathrm{ghl}}(\cdot, y_i)$ for $i = 1, ..., n$ and $z \in \mathbb{R}$

$$\begin{aligned}
-\big(v^u_{\mathrm{ghl}}(\cdot, y_i)\big)^*(z) &= \sup_{k\geq 0}\left\{\inf_{a\in\mathbb{R},\, t\geq 0}\left\{ - za + t^u + k(1 - ay_i - t)\right\}\right\} \\
&= \sup_{k\geq 0}\left\{\inf_{a\in\mathbb{R}}\{-za - kay_i\} + \inf_{t\geq 0}\{t^u - kt\} + k\right\} \\
&= \sup_{\substack{k\geq 0,\\ k=-zy_i}}\left\{(1 - u)\left(\frac{k}{u}\right)^{\frac{u}{u-1}} + k\right\} \\
&= \begin{cases} (1 - u)\left(\frac{-zy_i}{u}\right)^{\frac{u}{u-1}} - zy_i, & \text{if } zy_i \leq 0, \\ -\infty, & \text{otherwise.} \end{cases}
\end{aligned}$$

Hence, the corresponding dual problem to $(\mathrm{P}^u_{\mathrm{ghl}})$ looks like

$$(\mathrm{D}^u_{\mathrm{ghl}}) \qquad \sup_{\substack{P_i\in\mathbb{R},\\ P_iy_i\geq 0, i=1...,n}} \left\{\frac{1 - u}{(Cu^u)^{\frac{1}{u-1}}}\sum_{i=1}^n (P_iy_i)^{\frac{u}{u-1}} + \sum_{i=1}^n P_iy_i - \frac{1}{2}P^{\mathrm{T}}KP\right\}.$$

Formulated as an infimum problem, $(\mathrm{D}^u_{\mathrm{ghl}})$ becomes

$$(\mathrm{D}^u_{\mathrm{ghl}}) \qquad \inf_{\substack{P_i\in\mathbb{R},\\ P_iy_i\geq 0, i=1...,n}} \left\{\frac{1}{2}P^{\mathrm{T}}KP + \frac{u - 1}{(Cu^u)^{\frac{1}{u-1}}}\sum_{i=1}^n (P_iy_i)^{\frac{u}{u-1}} - \sum_{i=1}^n P_iy_i\right\},$$

while, by taking $\alpha = (\alpha_1, \ldots, \alpha_n)^{\mathrm{T}} \in \mathbb{R}^n$, $\alpha_i := P_iy_i$, $i = 1, \ldots, n$, one obtains for it the following equivalent formulation

$$(\mathrm{D}^u_{\mathrm{ghl}}) \qquad \inf_{\alpha_i\geq 0,\, i=1,...,n} \left\{\frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_jy_iy_jK_{ij} + \frac{u - 1}{(Cu^u)^{\frac{1}{u-1}}}\sum_{i=1}^n \alpha_i^{\frac{u}{u-1}} - \sum_{i=1}^n \alpha_i\right\}.$$

This problem gives rise for $u = 2$ to

$$(\mathrm{D}^2_{\mathrm{ghl}}) \qquad \inf_{\alpha_i\geq 0,\, i=1,...,n} \left\{\frac{1}{2}\left(\sum_{i,j=1}^n \alpha_i\alpha_jy_iy_jK_{ij} + \frac{1}{2C}\sum_{i=1}^n \alpha_i^2\right) - \sum_{i=1}^n \alpha_i\right\}$$

and for $u = 3$ to

$$(\mathrm{D}^3_{\mathrm{ghl}}) \qquad \inf_{\alpha_i \geq 0,\, i=1,\ldots,n} \left\{ \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K_{ij} + \frac{2}{\sqrt{27C}} \sum_{i=1}^{n} \alpha_i^{\frac{3}{2}} - \sum_{i=1}^{n} \alpha_i \right\},$$

which are the situations that we employ, along the one corresponding to the hinge loss, in Section 3.3 for solving the classification task.

Remark 3.8. The problems $(\mathrm{D}_{\mathrm{hl}})$ and $(\mathrm{D}^2_{\mathrm{ghl}})$ are convex quadratic optimization problems with affine inequality constraints and they can be solved by making use of one of the standard solvers which exist for this class of optimization problems. This is not anymore the case for $(\mathrm{D}^3_{\mathrm{ghl}})$, which is however a convex optimization problem. Thus one can use for solving it instead one of the standard solvers for convex differentiable optimization problems with affine inequality constraints. In order to solve both the quadratic and the non-quadratic optimization problems, we applied appropriate optimization routines from the MATLAB® optimization toolbox involving interior point methods for convex quadratic optimization problems (see for example [33, 25]).

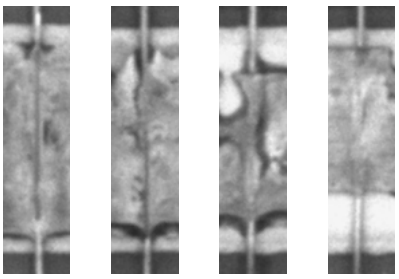
3.3 Application to image classification

In this section we describe the data for which the classification task, based on the approach described above, has been performed. Furthermore, we illustrate how the data has been preprocessed and give numerical results for the problems $(\mathrm{D}_{\mathrm{hl}})$, $(\mathrm{D}^2_{\mathrm{ghl}})$ and $(\mathrm{D}^3_{\mathrm{ghl}})$ arising when considering the different loss functions investigated in Section 3.2. These investigations can be extended by considering other loss functions and by calculating the corresponding dual problems. The only assumption we need for the former is convexity and lower semicontinuity in the first component, which the majority of the popular loss functions (with the exception of the $0-1$-*loss*) fulfill.

3.3.1 Training data

The available data were photographs of components used in the automotive industry, taken by a camera that is an internal part of the machine that produces these items. The overall task is to decide whether a produced component is fine or has to be considered as defective. In particular, a component is considered to be fine if a wire has been brazed correctly onto an attachment and it is defective otherwise. Consequently, a binary classification problem arises, where the label $+1$ denotes the class of components that are fine and the label -1 denotes the class of components that are defective. In other words, the goal of the classification task is to distinguish good joints from bad joints.

There was a total number of 4740 photographs of the components available, represented as gray scale images of size 200×50 pixels. Consisting of 2416 images of class $+1$ and 2324 images of class -1 the data set was nearly balanced. Since each pixel of the 8-bit gray-scale image represents a specific shade of gray, we assigned to it a value between 0 to 255, where the value equals 0 if the pixel is purely black and 255 if the pixel is purely white, respectively. Figure 3.1 shows four example images, two of each class.



(a) (b) (c) (d)

Figure 3.1: Example images of good ((a), (b)) and bad ((c), (d)) joints.

3.3.2 Preprocessing

In order to be able to use the images for the classification task, we first transformed them into vectors. First, each of the images has been represented as a matrix $M_t \in \mathbb{R}^{200\times 50}$, $M_t = (m^t_{i,j})^{200,50}_{i,j=1}$, $t = 1, \ldots, 4740$, with entries $m^t_{ij} \in \{0, 1, \ldots, 255\}$, $i = 1, \ldots, 200$, $j = 1, \ldots, 50$. By simply concatenating the rows of the matrix M_t, we obtained a vector m_t representing image t, i. e.

$$m_t = (m^t_{11}, \ldots, m^t_{1\,200}, \ldots, m^t_{50\,1}, \ldots, m^t_{50\,200})^T = (m_{t\,1}, \ldots, m_{t\,10\,000})^T \in \mathbb{R}^{10\,000}.$$

Denote by $\mathcal{D} = \{(m_t, y_t), t = 1, \ldots, 4740\} \subset \mathbb{R}^{10\,000} \times \{-1, +1\}$ the set of all data available. Following [48], the data has been normalized by dividing each data point by the quantity $(\frac{1}{4740}\sum_{t=1}^{4740} \|m_t\|^2)^{\frac{1}{2}}$, due to numerical reasons. Despite the fact that nowadays computations can in fact be performed for $10\,000-$dimensional vectors, we found it desirable to reduce their dimension to a dimension for which computations can be performed comparatively fast, especially concerning the calculation of the kernel matrix and the value of the decision function. For that reason, a so-called *feature ranking* was performed, by assigning a score to each pixel indicating its relevance for distinguishing between the two classes. Therefore, for the set of input data $D = \{m_1, \ldots, m_{4740}\}$ we defined the sets

$$D^+ := \{m_t \in D : y_t = +1\ \} \text{ and } D^- := \{m_t \in D : y_t = -1\}.$$

For both of these data sets, we calculated the mean μ_i,

$$\mu_i(D^+) = \frac{1}{|D^+|} \sum_{m_j \in D^+} m_{ji}, \quad \mu_i(D^-) = \frac{1}{|D^-|} \sum_{m_j \in D^-} m_{ji}, \quad i = 1, \ldots, 10\,000,$$

and the variance σ_i^2,

$$\sigma_i^2(D^+) = \frac{1}{|D^+|} \sum_{m_j \in D^+} (m_{ji} - \mu_i(D^+))^2,$$
$$\sigma_i^2(D^-) = \frac{1}{|D^-|} \sum_{m_j \in D^-} (m_{ji} - \mu_i(D^-))^2,$$

$i = 1, \ldots, 10\,000$, for each separate pixel of the images in the sets D^+ and D^-. The score S_i for the $i-$th pixel has been then calculated by

$$S_i(D) = \frac{(\mu_i(D^+) - \mu_i(D^-))^2}{\sigma_i^2(D^+) + \sigma_i^2(D^-)} \text{ for } i = 1, \ldots, 10\,000.$$

By applying this approach to the data set of images (cf. Figure (3.1)), we determined a score for each pixel, indicating its relevance for the classification task. Figure 3.2 plots the scores that have been assigned to the separate pixels. Finally, we have chosen only the pixels with a score greater or equal than 0.1 in order to reduce the dimension of the input data. This approach provided a number of 4398 pixel relevant for the classification task.

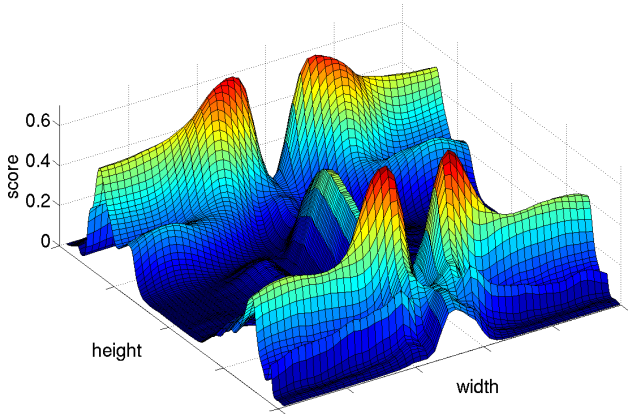


Figure 3.2: Visualization of the scores of the pixels.

3.3.3 Numerical results

To obtain a classifier numerical tests were performed for the three choices of the loss function discussed in the previous section, namely the hinge loss

$v_{\mathrm{hl}}(a,y) = (1-ay)_+$ and the generalized hinge loss $v^u_{\mathrm{ghl}}(a,y) = (1-ay)^u_+$ for $u = 2$ and $u = 3$ and the corresponding three dual problems (D_{hl}), ($\mathrm{D}^2_{\mathrm{ghl}}$) and, respectively, ($\mathrm{D}^3_{\mathrm{ghl}}$) were used. As kernel function the *Gaussian RBF kernel*

$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \tag{3.15}$$

with parameter $\sigma > 0$ was chosen. This gave rise to a positive definite Gram matrix K and, therefore, according to Remark 3.7, an optimal solution $\bar{P} := (\bar{P}_1, ..., \bar{P}_n)^T$ of the dual was an optimal solution of the primal, too. In this way the components of this vector provided the decision function we looked for when considering the classification task. Since the regularization parameter C and the kernel parameter σ were unknown and had to be determined by the user, first, a 10-fold cross validation was performed for each of the three loss functions and for each combination (C, σ) from a given set of values for each parameter. The whole

loss function	C	σ			
		0.1	0.5	1	10
hinge loss	1	0.2321	0.3376	0.4220	49.030
	10	0.1899	0.2321	0.3165	0.6752
	100	0.1899	0.1688	0.2532	0.4220
	1000	0.1899	0.2110	0.3587	0.2954
quadratic hinge loss	1	0.2110	0.2743	0.3376	2.1100
	10	0.2110	0.2110	0.2532	0.4642
	100	0.1899	0.1688	0.2954	0.3587
	1000	0.1899	0.2110	0.3165	0.3376
cubic hinge loss	1	0.2110	0.2532	0.2954	1.0972
	10	0.1899	0.2321	0.3376	0.4431
	100	0.1899	0.1899	0.3165	0.3376
	1000	0.1899	0.2110	0.3165	0.3587

Table 3.1: Average classification errors over ten folds in percentage of the number of images contained in the test sets.

data set was split into ten disjoint and equally sized subsets resulting in ten folds, each of them consisting of 474 input data points. The average classification error over all ten test folds for each parameter combination and for each loss function was computed, giving information about the corresponding best combination of parameters C and σ. Table 3.1 shows the average classification errors over ten folds for a selection of tested parameter combinations.

As one can see, the classification errors are remarkably small for all loss functions and for nearly all combinations of the kernel parameter σ and the regularization parameter C. There is an average misclassification rate of only up to 1% of the images contained in the test sets. The smallest errors occur for the combination $C = 100$ and $\sigma = 0.5$ for all loss functions. Taking this parameter combination as the optimal one, one obtains a number of 151 support vectors for the *hinge loss* function, i. e. only 3.2% of the images of the whole training data set are needed to fully describe the decision function. Concerning the *quadratic hinge loss*, we obtained 178 support vectors which is just a little more than for the usual hinge loss function. For the cubic hinge loss a total of 2207 support vectors was obtained, which is nearly the half of the full training set.

In order to compare the overall performance of the resulting classifier for different choices of the loss function we performed a nested cross validation (cf. [72, 59]), too. In this way one obtains an unbiased estimate of the true classification error for each model. More precisely, we implemented a so-called two nested 10-fold cross validation, i. e. for an outer loop the whole set of images was split into ten disjoint sets used as test sets to obtain the classification error. For each test set the remaining data again was split into ten disjoint sets used in the inner loop. On the basis of these ten sets the 10-fold cross validation described above was performed to determine both the optimal kernel parameter and the regularization parameter. Once these parameters are determined, they

loss function	hinge loss	quadratic hinge loss	cubic hinge loss
average test error	0.050	0.041	0.046

Table 3.2: The average misclassification rate obtained via the two nested 10-fold cross validation for each loss function.

were used for training a classifier on the whole set and for testing it on the remaining test set. As result we got the average test error over ten test sets for each choice of the loss function, which provided a characteristic for comparing the performance of these classifiers. The results are shown in Table 3.2 and emphasize that the quadratic hinge loss function works best for this image classification task.

3.4 Dual programs for the regression task

In this section we consider different loss functions for performing the regression task. For each of the regularization problems to which these loss functions give rise we derive the corresponding dual problem. Notice also that all considered

loss functions in this section are proper, convex and lower semicontinuous in their first arguments.

3.4.1 The $\varepsilon-$insensitive loss function

The well known ε-insensitive loss function $v_\varepsilon : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is, for $\varepsilon > 0$, defined as

$$v_\varepsilon(a, y) = |a - y|_\varepsilon := (|a - y| - \varepsilon)_+ = \begin{cases} 0, & |a - y| \leq \varepsilon, \\ |a - y| - \varepsilon, & \text{else.} \end{cases} \tag{3.16}$$

Thus the primal optimization problem $\left(\bar{\mathrm{P}}_{\mathrm{SV}}\right)$ becomes

$$(\mathrm{P}_\varepsilon) \qquad \inf_{c \in \mathbb{R}^n} \left\{ C \sum_{i=1}^{n} (|(Kc)_i - y_i| - \varepsilon)_+ + \frac{1}{2} c^{\mathrm{T}} K c \right\}. \tag{3.17}$$

To obtain its dual problem (D_ε) as a particular instance of $\left(\bar{\mathrm{D}}_{\mathrm{SV}}\right)$ we use the Lagrange technique in order to calculate the conjugate function of $v_\varepsilon(\cdot, y_i)$, for $i = 1, ..., n$. For $z \in \mathbb{R}$ and $y \in \mathbb{R}$ we have

$$\begin{aligned} -\left(v_\varepsilon(\cdot, y)\right)^*(z) &= -\sup_{a \in \mathbb{R}} \{za - (|a - y| - \varepsilon)_+\} = \inf_{a \in \mathbb{R}} \{-za + (|a - y| - \varepsilon)_+\} \\ &= \inf_{\substack{a \in \mathbb{R}, \\ t \geq 0,\, t \geq |a-y| - \varepsilon}} \{-za + t\} \\ &= \sup_{\lambda \geq 0,\, \beta \geq 0} \left\{ \inf_{a \in \mathbb{R},\, t \in \mathbb{R}} \{-za + t + \lambda|a - y| - \lambda\varepsilon - \lambda t - \beta t\} \right\} \\ &= \sup_{\lambda \geq 0,\, \beta \geq 0} \left\{ \inf_{a \in \mathbb{R}} \{-za + \lambda|a - y|\} + \inf_{t \in \mathbb{R}} \{t - \lambda t - \beta t\} - \lambda\varepsilon \right\}. \end{aligned}$$

Since

$$\inf_{a \in \mathbb{R}} \{-za + \lambda|a - y|\} = \begin{cases} -zy, & \lambda \geq |z|, \\ -\infty, & \text{else} \end{cases}$$

and

$$\inf_{t \in \mathbb{R}} \{t - \lambda t - \beta t\} = \begin{cases} 0, & \lambda + \beta = 1, \\ -\infty, & \text{else}, \end{cases}$$

we get

$$-\left(v_\varepsilon(\cdot, y)\right)^*(z) = \begin{cases} -zy - \varepsilon|z|, & |z| \leq 1, \\ -\infty, & \text{else} \end{cases}$$

and the dual problem (D_ε) to the primal problem (P_ε) results in

$$(\mathrm{D}_\varepsilon) \qquad \sup_{\substack{P=(P_1,\ldots,P_n)^{\mathrm{T}}\in\mathbb{R}^n,\\ |P_i|\leq C, i=1,\ldots,n}} \left\{ \sum_{i=1}^n P_i y_i - \varepsilon \sum_{i=1}^n |P_i| - \frac{1}{2} P^{\mathrm{T}} K P \right\}. \tag{3.18}$$

3.4.2 The quadratic ε-insensitive loss function

The second loss function we consider here is the so-called quadratic $\varepsilon-$insensitive loss function $v_{\varepsilon^2} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, which is defined, for $\varepsilon > 0$, by

$$v_{\varepsilon^2}(a,y) = (|a-y|_\varepsilon)^2 = (|a-y|-\varepsilon)_+^2 = \begin{cases} 0, & |a-y| \leq \varepsilon, \\ (|a-y|-\varepsilon)^2, & \text{else.} \end{cases} \tag{3.19}$$

The corresponding primal problem reads in this case

$$(\mathrm{P}_{\varepsilon^2}) \qquad \inf_{c\in\mathbb{R}^n} \left\{ C \sum_{i=1}^n (|(Kc)_i - y_i| - \varepsilon)_+^2 + \frac{1}{2} c^{\mathrm{T}} K c \right\}.$$

Again, in order to derive its dual problem $(\mathrm{D}_{\varepsilon^2})$, we need to calculate, for $y, z \in \mathbb{R}$, the following conjugate function

$$\begin{aligned} -\left(v_{\varepsilon^2}(\cdot,y)\right)^*(z) &= -\sup_{a\in\mathbb{R}} \{za - (|a-y|-\varepsilon)_+^2\} = \inf_{\substack{a\in\mathbb{R},\\ t\geq 0,\, t\geq |a-y|-\varepsilon}} \{-za + t^2\} \\ &= \sup_{\lambda\geq 0,\,\beta\geq 0} \left\{ \inf_{a\in\mathbb{R},\, t\in\mathbb{R}} \left\{ -za + t^2 + \lambda(|a-y|-\varepsilon-t) - \beta t \right\} \right\} \\ &= \sup_{\lambda\geq 0,\,\beta\geq 0} \left\{ \inf_{a\in\mathbb{R}} \{-za + \lambda|a-y|\} + \inf_{t\in\mathbb{R}} \left\{ t^2 - \lambda t - \beta t \right\} - \lambda\varepsilon \right\}. \end{aligned}$$

The first inner infimum has been already calculated in the previous subsection, while for the second one we have

$$\inf_{t\in\mathbb{R}} \left\{ t^2 - (\lambda+\beta)t \right\} = -\frac{1}{4}(\lambda+\beta)^2.$$

Hence, the above conjugate becomes

$$-\left(v_{\varepsilon^2}(\cdot,y)\right)^*(z) = -zy - \frac{1}{4}z^2 - \varepsilon|z|$$

and gives rise to the following dual problem

$$(\mathrm{D}_{\varepsilon^2}) \qquad \sup_{P=(P_1,\ldots,P_n)^{\mathrm{T}}\in\mathbb{R}^n} \left\{ \sum_{i=1}^n P_i y_i - \frac{1}{4C} \sum_{i=1}^n P_i^2 - \varepsilon \sum_{i=1}^n |P_i| - \frac{1}{2} P^{\mathrm{T}} K P \right\}. \tag{3.20}$$

3.4.3 The Huber loss function

Another popular choice for the loss function in SVM regression tasks is the Huber loss function $v_{\mathrm{H}} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ introduced in [43] which is defined, for $\varepsilon > 0$, as

$$v_{\mathrm{H}}(a, y) = \begin{cases} \varepsilon|a - y| - \frac{\varepsilon^2}{2}, & |a - y| > \varepsilon, \\ \frac{1}{2}|a - y|^2, & |a - y| \leq \varepsilon. \end{cases} \tag{3.21}$$

The primal problem associated with the Huber loss function therefore becomes

$$(\mathrm{P_H}) \qquad \inf_{c \in \mathbb{R}^n} \left\{ C \sum_{i=1}^{n} v_{\mathrm{H}}((Kc)_i, y_i) + \frac{1}{2} c^{\mathrm{T}} K c \right\}.$$

For all $z, y \in \mathbb{R}$ one has

$$\begin{aligned} -\left(v_{\mathrm{H}}(\cdot, y)\right)^*(z) &= -\sup_{a \in \mathbb{R}} \{za - v_{\mathrm{H}}(a, y)\} = \inf_{a \in \mathbb{R}} \{-za + v_{\mathrm{H}}(a, y)\} \\ &= \min \left\{ \inf_{\substack{a \in \mathbb{R}, \\ |a-y| \leq \varepsilon}} \left\{ -za + \frac{1}{2}|a - y|^2 \right\}, \inf_{\substack{a \in \mathbb{R}, \\ |a-y| > \varepsilon}} \left\{ -za + \varepsilon|a - y| - \frac{\varepsilon^2}{2} \right\} \right\} \\ &= \min \left\{ \inf_{\substack{a \in \mathbb{R}, \\ |a-y| \leq \varepsilon}} \left\{ -za + \frac{1}{2}(a - y)^2 \right\}, \inf_{\substack{a \in \mathbb{R}, \\ a > y + \varepsilon}} \left\{ -za + \varepsilon(a - y) - \frac{\varepsilon^2}{2} \right\}, \right. \\ &\qquad \left. \inf_{\substack{a \in \mathbb{R}, \\ a < y - \varepsilon}} \left\{ -za + \varepsilon(y - a) - \frac{\varepsilon^2}{2} \right\} \right\}. \end{aligned}$$

For the first infimum we get

$$\inf_{\substack{a \in \mathbb{R}, \\ |a-y| \leq \varepsilon}} \left\{ -za + \frac{1}{2}(a - y)^2 \right\} = \begin{cases} \frac{\varepsilon^2}{2} - zy + z\varepsilon - \frac{y^2}{2}, & z < -\varepsilon, \\ -\frac{1}{2}z^2 - zy - \frac{y^2}{2}, & z \in [-\varepsilon, \varepsilon], \\ \frac{\varepsilon^2}{2} - zy - z\varepsilon - \frac{y^2}{2}, & z > \varepsilon, \end{cases} \tag{3.22}$$

while the second and third infima result in

$$\inf_{\substack{a \in \mathbb{R}, \\ a > y + \varepsilon}} \left\{ -za + \varepsilon(a - y) - \frac{\varepsilon^2}{2} \right\} = \begin{cases} \frac{\varepsilon^2}{2} - zy - z\varepsilon, & z \leq \varepsilon, \\ -\infty, & \text{else} \end{cases} \tag{3.23}$$

and

$$\inf_{\substack{a \in \mathbb{R}, \\ a < y - \varepsilon}} \left\{ -za + \varepsilon(y - a) - \frac{\varepsilon^2}{2} \right\} = \begin{cases} \frac{1}{2}\varepsilon^2 - zy + z\varepsilon, & z \geq -\varepsilon, \\ -\infty, & \text{else}, \end{cases} \tag{3.24}$$

respectively. Putting (3.22), (3.23) and (3.24) together we obtain the following formula for the conjugate function

$$-\left(v_{\mathrm{H}}(\cdot,y)\right)^*(z) = \begin{cases} \min\left\{-\frac{1}{2}z^2 - zy, \frac{\varepsilon^2}{2} - zy - z\varepsilon, \frac{\varepsilon^2}{2} - zy + \varepsilon z\right\}, & z \in [-\varepsilon,\varepsilon], \\ -\infty, & \text{else}, \end{cases}$$
$$= \begin{cases} -\frac{1}{2}z^2 - zy, & z \in [-\varepsilon,\varepsilon], \\ -\infty, & \text{else}. \end{cases}$$

Thus, the dual problem to $(\mathrm{P_H})$ reads

$$(\mathrm{D_H}) \qquad \sup_{\substack{P=(P_1,\ldots,P_n)^{\mathrm{T}}\in\mathbb{R}^n,\\ |P_i|\leq\varepsilon C, i=1,\ldots,n}} \left\{\sum_{i=1}^n P_i y_i - \frac{1}{2C}\sum_{i=1}^n P_i^2 - \frac{1}{2}P^{\mathrm{T}}KP\right\}.$$

3.4.4 The extended loss function

Finally, we provide the resulting dual problem when using the extended loss function $v_{\mathrm{ext}} : \mathbb{R}\times\mathbb{R}\to\overline{\mathbb{R}}$, which is defined, for $\varepsilon > 0$, as

$$v_{\mathrm{ext}}(a,y) = \delta_{[-\varepsilon,\varepsilon]}(a-y) = \begin{cases} 0, & |a-y|\leq\varepsilon, \\ +\infty, & \text{else}. \end{cases} \tag{3.25}$$

This choice gives rise to the following primal problem

$$(\mathrm{P_{ext}}) \qquad \inf_{\substack{c\in\mathbb{R}^n,\\ |(Kc)_i - y_i|\leq\varepsilon, i=1,\ldots,n}} \left\{\frac{1}{2}c^{\mathrm{T}}Kc\right\}.$$

By making again use of Lagrange duality, we get for all $y, z \in \mathbb{R}$

$$-\left(v_{\mathrm{ext}}(\cdot,y)\right)^*(z) = -\sup_{\substack{a\in\mathbb{R},\\ |a-y|\leq\varepsilon}}\{za\} = \inf_{\substack{a\in\mathbb{R},\\ |a-y|\leq\varepsilon}}\{-za\} = \inf_{\substack{a\in\mathbb{R},\\ a-y-\varepsilon\leq 0,\\ y-a-\varepsilon\leq 0}}\{-za\}$$
$$= \sup_{\lambda\geq 0,\,\beta\geq 0}\left\{\inf_{a\in\mathbb{R}}\{(-z+\lambda-\beta)a - \lambda y - \lambda\varepsilon + \beta y - \beta\varepsilon\}\right\}$$
$$= \sup_{\substack{\lambda\geq 0,\,\beta\geq 0,\\ \lambda-\beta=z}}\{-\lambda y - \lambda\varepsilon + \beta y - \beta\varepsilon\}$$
$$= \sup_{\substack{\lambda\geq 0,\,\beta\geq 0,\\ \lambda-\beta=z}}\{-(\lambda-\beta)y - (\lambda+\beta)\varepsilon\}$$
$$= -zy + \sup_{\substack{\lambda\geq 0,\,\beta\geq 0,\\ \lambda-\beta=z}}\{-\varepsilon(\lambda+\beta)\} = -zy - \varepsilon|z|.$$

Consequently, the dual problem to ($\mathrm{P_{ext}}$) has the following formulation

$$(\mathrm{D_{ext}}) \qquad \sup_{P=(P_1,\ldots,P_n)^{\mathrm{T}}\in\mathbb{R}^n} \left\{ \sum_{i=1}^{n} P_i y_i - \varepsilon \sum_{i=1}^{n} |P_i| - \frac{1}{2} P^{\mathrm{T}} K P \right\}.$$

3.5 Application to regression tasks

In this section we discuss two particular regression tasks in the light of the approach introduced in the previous sections and solve to this end the different dual optimization problems (D_ε), ($\mathrm{D}_{\varepsilon^2}$), ($\mathrm{D_H}$) and ($\mathrm{D_{ext}}$) numerically. The accuracy of the regression will be compared using two data sets widely used as benchmarks in the literature (cf. [31, 60, 61]). In a first step, we reformulate these optimization problems in order to get a representation of them that is suitable for standard optimization routines and therefore more easy to handle with. Having in mind the dual problem (D_ε), we note that for $z \in \mathbb{R}$ it holds

$$|z| = \inf_{\substack{\alpha\geq 0,\alpha^*\geq 0,\\ \alpha-\alpha^*=z}} \{\alpha + \alpha^*\} \tag{3.26}$$

for arbitrary $z \in \mathbb{R}$. If $z \geq 0$, then the optimal solution of this minimization problem is $(\alpha, \alpha^*) = (z, 0)$, while, when $z < 0$, the optimal solution is $(\alpha, \alpha^*) = (0, -z)$. This remark constitutes the starting point for giving an equivalent formulation of the dual problem (D_ε) in terms of the variables α_i and α_i^*, $i = 1, \ldots, n$, which we will denote by ($\mathrm{D}_\varepsilon^\alpha$). For the problem

$$(\mathrm{D}_\varepsilon) \qquad \sup_{\substack{P=(P_1,\ldots,P_n)^{\mathrm{T}}\in\mathbb{R}^n,\\ |P_i|\leq C, i=1,\ldots,n}} \left\{ \sum_{i=1}^{n} P_i y_i - \varepsilon \sum_{i=1}^{n} |P_i| - \frac{1}{2} P^{\mathrm{T}} K P \right\}$$

the equivalent formulation ($\mathrm{D}_\varepsilon^\alpha$) is

$$(\mathrm{D}_\varepsilon^\alpha) \qquad \inf_{\substack{\alpha_i,\alpha_i^*\in[0,C],\\ i=1,\ldots,n}} \left\{ \frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K_{ij} + \varepsilon \sum_{i=1}^{n} (\alpha_i + \alpha_i^*) - \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) y_i \right\}.$$

Using again (3.26) an equivalent formulation for the problem

$$(\mathrm{D_{ext}}) \qquad \sup_{P=(P_1,\ldots,P_n)^{\mathrm{T}}\in\mathbb{R}^n} \left\{ \sum_{i=1}^{n} P_i y_i - \varepsilon \sum_{i=1}^{n} |P_i| - \frac{1}{2} P^{\mathrm{T}} K P \right\},$$

to which the use of extended loss gives rise, in terms of α_i and α_i^*, $i = 1, \ldots, n$, is

$$(\mathrm{D}_{\mathrm{ext}}^{\alpha}) \qquad \inf_{\substack{\alpha_i, \alpha_i^* \geq 0, \\ i=1,\ldots,n}} \left\{ \frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K_{ij} + \varepsilon \sum_{i=1}^{n} (\alpha_i + \alpha_i^*) - \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) y_i \right\}.$$

In order to obtain an equivalent formulation $(\mathrm{D}_{\varepsilon^2}^{\alpha})$ of the optimization problem $(\mathrm{D}_{\varepsilon^2})$ we make use of the fact that

$$|z| = \inf_{\substack{\alpha, \alpha^* \geq 0, \\ \alpha - \alpha^* = z}} \left\{ \alpha + \alpha^* + \frac{\alpha \alpha^*}{2C\varepsilon} \right\}$$

for arbitrary $z \in \mathbb{R}$. Then the representation of

$$(\mathrm{D}_{\varepsilon^2}) \qquad \sup_{P=(P_1,\ldots,P_n)^{\mathrm{T}} \in \mathbb{R}^n} \left\{ \sum_{i=1}^{n} P_i y_i - \frac{1}{4C} \sum_{i=1}^{n} P_i^2 - \varepsilon \sum_{i=1}^{n} |P_i| - \frac{1}{2} P^{\mathrm{T}} K P \right\}$$

is

$$(\mathrm{D}_{\varepsilon^2}^{\alpha}) \qquad \inf_{\alpha_i, \alpha_i^* \geq 0} \left\{ \frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K_{ij} + \frac{1}{4C} \sum_{i=1}^{n} (\alpha_i^2 + (\alpha_i^*)^2) + \varepsilon \sum_{i=1}^{n} (\alpha_i + \alpha_i^*) - \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) y_i \right\}.$$

Finally, for arbitrary $z \in \mathbb{R}$ it holds

$$z^2 = \inf_{\substack{\alpha, \alpha^* \geq 0, \\ \alpha - \alpha^* = z}} \{\alpha^2 + (\alpha^*)^2\}$$

and therefore, an equivalent formulation of

$$(\mathrm{D}_{\mathrm{H}}) \qquad \sup_{\substack{P=(P_1,\ldots,P_n)^{\mathrm{T}} \in \mathbb{R}^n, \\ |P_i| \leq \varepsilon C, i=1,\ldots,n}} \left\{ \sum_{i=1}^{n} P_i y_i - \frac{1}{2C} \sum_{i=1}^{n} P_i^2 - \frac{1}{2} P^{\mathrm{T}} K P \right\}$$

in terms of α_i and α_i^*, $i = 1, \ldots, n$, is

$$(\mathrm{D}_{\mathrm{H}}^{\alpha}) \inf_{\substack{\alpha_i, \alpha_i^* \in [0, \varepsilon C], \\ i=1,\ldots,n}} \left\{ \frac{1}{2} \sum_{i,j=1}^{n} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K_{ij} + \frac{1}{2C} \sum_{i=1}^{n} \left(\alpha_i^2 + (\alpha_i^*)^2\right) - \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) y_i \right\}.$$

Remark 3.9. While the corresponding primal problems are either unconstrained nondifferentiable convex optimization problems or reformulations of constrained optimization problems with differentiable objective functions and not easily handleable inequality constraints, the duals $(\mathrm{D}_{\varepsilon}^{\alpha})$, $(\mathrm{D}_{\varepsilon^2}^{\alpha})$, $(\mathrm{D}_{\mathrm{ext}}^{\alpha})$ and $(\mathrm{D}_{\mathrm{H}}^{\alpha})$ assume the minimization of a convex quadratic objective function over some feasible sets expressed via box constraints or nonnegative orthants. This makes them easier solvable via some standard algorithms designed for these classes of optimization problems than their corresponding primal problems. Moreover, if $(\bar{\alpha}_1, \bar{\alpha}_1^*, ..., \bar{\alpha}_n, \bar{\alpha}_n^*)$ represents an optimal solution of each of the reformulated dual problems, then $\bar{P} := (\bar{P}_1, ..., \bar{P}_n)^{\mathrm{T}}, \bar{P}_i = \bar{\alpha}_i - \bar{\alpha}_i^*$, $i = 1, \ldots, n$, represents an optimal solution of the corresponding initial dual.

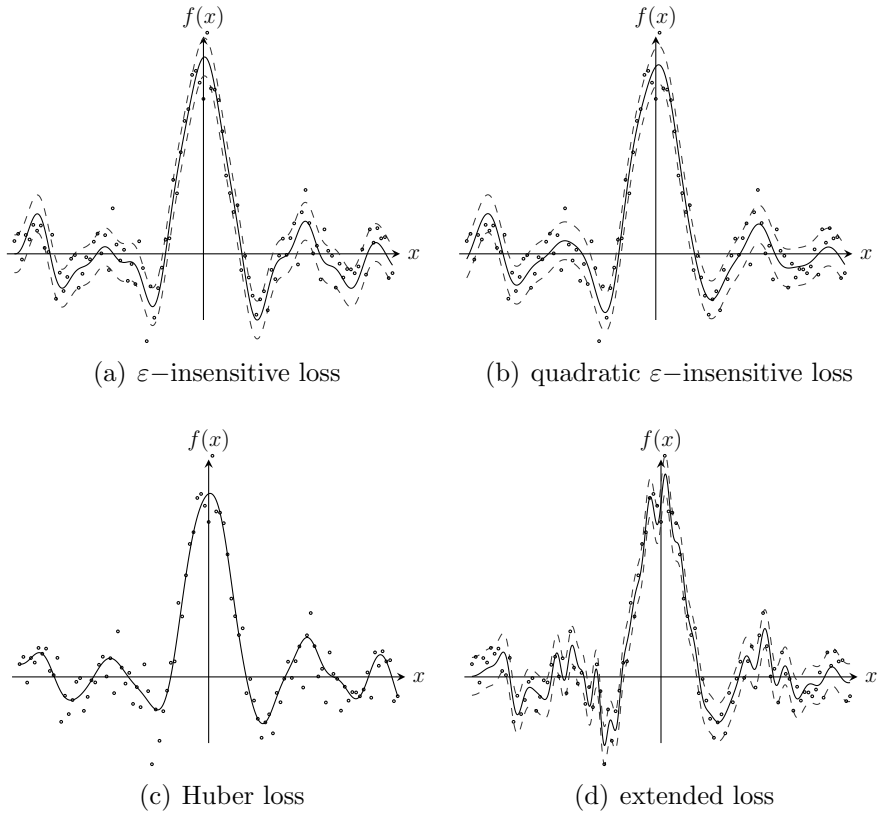
The two particular regression tasks which we consider in this section involve a toy data set (cf. 3.5.1) and the popular Boston Housing data set (cf. 3.5.2). In both situations we use the Gaussian RBF kernel (3.15) with kernel parameter $\sigma > 0$. This gives rise to a positive definite Gram matrix K and, therefore, according to Remark 3.7, an optimal solution $\bar{P} := (\bar{P}_1, ..., \bar{P}_n)^{\mathrm{T}}$ of the dual will be an optimal solution of the primal, too. Thus, the components of this vector will provide the decision function one looks for when considering the regression task.

3.5.1 A toy data set

In this subsection we numerically solve a regression task where the data has been sampled from the function $f : \mathbb{R} \to \mathbb{R}$,

$$f(x) = \begin{cases} \frac{\sin(x)}{x}, & x \neq 0, \\ 1, & x = 0. \end{cases}$$

The function values for all $x \in X = \{-5.0, -4.9, \ldots, 4.9, 5.0\}$ resulting in a total of 101 points were sampled. The values $f(x)$, $x \in X$, were perturbed by adding a random value drawn from the normal distribution $\mathcal{N}(0, 0.1)$. In this way a training set $\mathcal{D} = \{(x_i, y_i) : i = 1, \ldots, 101\}$ was obtained and used for training. On the basis of this set we solved the dual problems $(\mathrm{D}_{\varepsilon}^{\alpha})$, $(\mathrm{D}_{\varepsilon^2}^{\alpha})$, $(\mathrm{D}_{\mathrm{H}}^{\alpha})$ and $(\mathrm{D}_{\mathrm{ext}}^{\alpha})$ numerically, while Figure 3.3 shows the shapes of the resulting regression functions when choosing the corresponding loss function.



(a) $\varepsilon-$insensitive loss (b) quadratic $\varepsilon-$insensitive loss

(c) Huber loss (d) extended loss

Figure 3.3: Illustrations of the four resulting regression functions (solid lines) for the corresponding loss function and the $\varepsilon-$tube (dashed lines, where appropriate) based on the generated training set (dots). (a) $C = 100$, $\sigma = 0.5$, $\varepsilon = 0.1$ (b) $C = 100$, $\sigma = 0.5$, $\varepsilon = 0.1$ (c) $C = 100$, $\sigma = 0.5$, $\varepsilon = 0.1$ (d) $\sigma = 0.2$, $\varepsilon = 0.1$

Table 3.3 shows the corresponding mean squared errors. With respect to this special setting the use of the $\varepsilon-$insensitive loss function and of the quadratic $\varepsilon-$insensitive loss function produce similar mean squared errors, while the use of the extended loss function provides the lowest mean squared error, as expected.

loss function	$\varepsilon-$insensitive	ε^2-insensitive	Huber	extended
mean squared error	0.008192	0.008193	0.007566	0.006188

Table 3.3: The mean squared error for the four different loss functions obtained by applying the parameter settings described in the caption of Figure 3.3.

3.5.2 Boston Housing data set

In this section we solve the dual problems (D_{ε}^{α}), ($D_{\varepsilon^2}^{\alpha}$), ($D_{H}^{\alpha}$) and ($D_{\text{ext}}^{\alpha}$) for the the well known Boston Housing data set. This data set consists of 506 instances each of them described by 13 attributes. For a detailed description of the data

		ε		
C	σ	0.01	0.15	1.0
10	0.1	39.60	41.69	61.78
	0.5	10.51	9.21	34.37
	1.0	11.83	11.55	26.50
100	0.1	39.60	41.69	61.78
	0.5	12.58	10.54	34.13
	1.0	10.37	9.46	26.93
1000	0.1	39.60	41.69	61.78
	0.5	26.48	14.66	34.13
	1.0	15.45	10.16	26.93

(a) $\varepsilon-$insensitive loss

		ε		
C	σ	0.01	0.15	1.0
10	0.1	40.14	42.39	62.42
	0.5	8.63	9.30	36.52
	1.0	9.79	10.50	31.92
100	0.1	39.64	41.76	61.85
	0.5	10.37	9.77	34.39
	1.0	8.33	8.85	27.51
1000	0.1	39.60	41.69	61.79
	0.5	17.03	11.96	34.16
	1.0	10.49	9.85	26.99

(b) quadratic $\varepsilon-$insensitive loss

		ε		
C	σ	0.01	0.15	1.0
10	0.1	72.96	43.66	40.03
	0.5	33.79	13.72	8.95
	1.0	34.19	15.85	10.89
100	0.1	47.27	39.55	39.55
	0.5	15.61	10.18	10.02
	1.0	17.32	10.62	8.67
1000	0.1	39.52	39.52	39.52
	0.5	10.61	13.38	17.56
	1.0	11.89	10.24	10.13

(c) Huber loss

	ε			
σ	0.01	0.1	0.15	0.25
0.1	39.60	40.84	41.69	43.60
0.2	17.59	16.81	17.15	18.65
0.3	19.15	14.00	13.07	13.14
0.5	48.88	22.47	17.38	11.64
1.0	151.33	79.68	48.66	20.22
2.0	530.87	254.01	147.94	39.83

(d) extended loss

Figure 3.4: Four tables representing the average mean squared error over ten test folds for the resulting regression functions w. r. t. the corresponding loss functions and different parameter combinations.

set we refer to [76]. In order to determine good parameter choices for the kernel parameter σ, the regularization parameter C and the loss function parameter ε, we performed a 10-fold cross validation. In tables 3.4(a), 3.4(b), 3.4(c) and

3.4(d) the mean test errors over 10 folds for all four loss functions are shown for a part of the whole tested parameter values, where we choose the mean squared error for evaluation. As in [66], we scaled the data before solving the problems numerically. As one can notice, the best result, i. e. the lowest mean squared error over 10 test folds, is obtained for the quadratic $\varepsilon-$insensitive loss function followed by the $\varepsilon-$insensitive loss function and the Huber loss function.

Chapter 4

Double smoothing technique for a general optimization problem

In this chapter we will develop a framework for solving the general optimization problem of minimizing the sum of a function and a sum of m functions each composed with a linear operator introduced in Subsection 2.2.2. The considered optimization problem is an unconstrained and convex problem in finite dimensions which is in general not differentiable. A Fenchel-type dual problem will be assigned to it which can be obtained by making use of the perturbation approach (cf. [17, 12, 77, 37]). In that case weak duality always holds and if, in addition to it, even strong duality holds it is common to solve the dual problem instead of the primal one to obtain an optimal primal solution.

In general, in an appropriate setting, one can use methods like steepest descent methods, Newton's method or fast gradient methods (cf. [9, 50, 27]) to solve these problems. If, on the other hand, the problem is an unconstrained, convex and non-differentiable optimization problem one could make use of subgradient methods (see [50]). The aim of this chapter is to develop an efficient algorithm for solving our general optimization problem that is, compared to subgradient methods, faster in convergence since these methods can not achieve a rate of convergence better than $O(\frac{1}{\varepsilon^2})$, where $\varepsilon > 0$ is the accuracy for the objective value. We will instead solve the dual problem efficiently with a rate of convergence of $O\left(\frac{1}{\varepsilon}\ln\left(\frac{1}{\varepsilon}\right)\right)$ and construct an approximate primal solution by means of sequences that are generated by the algorithmic scheme on the basis of the dual iterates.

In particular, the main idea for the so-called double smoothing technique we will apply was employed in [34] and [35] for a special class of convex constrained optimization problems while the smoothing technique was introduced in [51, 52, 53]. To be able to apply the fast gradient method (see 2.3) efficiently we will regularize the dual problem, which is a non-smooth optimization problem, to obtain a strongly convex and continuously differentiable objective function with Lipschitz continuous gradient. This regularized dual problem is then solved via

the fast gradient method and we obtain a sequence of dual variables which in fact solve the non-smooth original dual problem. Furthermore, for this sequence we obtain convergence of the norm of the gradient of the single smoothed problem to zero, which is necessary to construct an approximate primal solution via the dual solution.

4.1 Problem formulation

Throughout this chapter we consider the optimization problem

$$(\mathrm{P}_{\mathrm{gen}}) \qquad \inf_{x\in\mathbb{R}^n}\left\{f(x)+\sum_{i=1}^{m} g_i(K_i x)\right\}$$

where $K_i : \mathbb{R}^n \to \mathbb{R}^{k_i}$, $i = 1,\dots,m$, are linear operators and the functions $g_i : \mathbb{R}^{k_i} \to \overline{\mathbb{R}}$ and $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ are assumed to be proper, convex and lower semicontinuous for $i = 1,\dots,m$. We further assume that the effective domains $\operatorname{dom} f \subset \mathbb{R}^n$ and $\operatorname{dom} g_i \subset \mathbb{R}^{k_i}$, $i = 1,\dots,m$, are nonempty and bounded sets. In order to formulate a Fenchel-type dual problem to $(\mathrm{P}_{\mathrm{gen}})$ we introduce the function $g : \mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_m} \to \overline{\mathbb{R}}$, $g(y_1,\dots,y_m) = \sum_{i=1}^m g_i(y_i)$ and the operator $K : \mathbb{R}^n \to \mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_m}$ via $Kx = (K_1x,\dots,K_mx)$. Then, the primal problem $(\mathrm{P}_{\mathrm{gen}})$ can be written as

$$\inf_{x\in\mathbb{R}^n}\left\{f(x)+g(Kx)\right\}, \tag{4.1}$$

where we assume $K(\operatorname{dom} f)\cap\operatorname{dom} g \neq \emptyset$ in order to deal with a non-degenerate optimization problem. To this optimization problem we assign the Fenchel-type dual problem

$$\sup_{(p_1,\dots,p_m)\in\mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_m}}\left\{-f^*\left(-K^*(p_1,\dots,p_m)\right)-g^*\left(p_1,\dots,p_m\right)\right\}, \tag{4.2}$$

following [12, 17, 77, 58], where $K^* : \mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_m} \to \mathbb{R}^n$, $K^*(y_1,\dots,y_m) = \sum_{i=1}^m K_i^*y_i$ with $K_i^* : \mathbb{R}^{k_i} \to \mathbb{R}^n$ the adjoint operator of K_i, $i = 1,\dots,m$, denotes the adjoint operator of K. The functions $g^* : \mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_m} \to \overline{\mathbb{R}}$ and $f^* : \mathbb{R}^n \to \overline{\mathbb{R}}$ are the conjugate functions of g and f, respectively, where $g_i^* : \mathbb{R}^{k_i} \to \overline{\mathbb{R}}$ are the conjugate functions of g_i, $i = 1,\dots,m$, and it holds $g^*(p_1,\dots,p_m) = \sum_{i=1}^m g_i^*(p_i)$ (cf. [17, Proposition 2.3.2]). This dual can be reformulated as

$$(\mathrm{D}_{\mathrm{gen}}) \qquad \sup_{(p_1,\dots,p_m)\in\mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_m}}\left\{-f^*\left(-\sum_{i=1}^{m}K_i^*p_i\right)-\sum_{i=1}^{m}g_i^*\left(p_i\right)\right\}.$$

As mentioned, weak duality between $(\mathrm{P}_{\mathrm{gen}})-(\mathrm{D}_{\mathrm{gen}})$ always holds by construction of the dual problem via the perturbation approach (cf. [17, 12, 77, 37]). We next investigate further properties for this primal-dual pair.

4.2 Existence of an optimal solution

Let us denote by $v\,(\mathrm{P_{gen}})$ and $v\,(\mathrm{D_{gen}})$ the optimal objective value of the primal problem $(\mathrm{P_{gen}})$ and the dual problem $(\mathrm{D_{gen}})$, respectively. In this section we will show that the primal problem $(\mathrm{P_{gen}})$ always has an optimal solution and that the optimal objective values coincide, i. e. $v\,(\mathrm{P_{gen}}) = v\,(\mathrm{D_{gen}})$. The main result of this section is stated in the next theorem.

Theorem 4.1. *For the primal-dual pair (P_{gen}) and (D_{gen}) introduced in Section 4.1 it holds $v\,(P_{\mathrm{gen}}) = v\,(D_{\mathrm{gen}})$. Furthermore, the primal problem (P_{gen}) has an optimal solution.*

For the proof of the above statement we will need the following lemma which is proven immediately after it is stated.

Lemma 4.2. *Let $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be a proper and lower semicontinuous function and $\operatorname{dom} f$ a bounded set. Then $\operatorname{dom} f^* = \mathbb{R}^n$.*

Proof. We have to show that $f^*(y) < \infty$ for all $y \in \mathbb{R}^n$. It holds

$$\operatorname{dom} f^* = \{y \in \mathbb{R}^n : f^*(y) < \infty\} = \left\{y \in \mathbb{R}^n : \sup_{x \in \operatorname{dom} f} \{\langle x, y\rangle - f(x)\} < \infty\right\}.$$

We further have

$$\sup_{x \in \operatorname{dom} f} \{\langle x, y\rangle - f(x)\} = -\inf_{x \in \operatorname{dom} f} \{f(x) - \langle x, y\rangle\}$$

and consider the optimization problem $\inf_{x \in \operatorname{dom} f}\{f(x) - \langle x, y\rangle\}$. Define, for each $y \in \mathbb{R}^n$, the function $h_y : \mathbb{R}^n \to \overline{\mathbb{R}}$, $h_y(x) := f(x) - \langle y, x\rangle$. Since f is proper and lower semicontinuous the function h_y is proper and lower semicontinuous and $\operatorname{dom} h_y = \operatorname{dom} f$. We now have to consider the problem $\inf_{x \in \operatorname{dom} f}\{h_y(x)\}$. Since h_y is proper there exists a $z \in \operatorname{dom} f$ such that $h_y(z) < \infty$. Furthermore, $h_y^* := \inf_{x \in \operatorname{dom} f}\{h_y(x)\} \leq h_y(z)$.

Assume now that $h_y^* = -\infty$. Let $(x_k)_{k \geq 1} \subseteq \operatorname{dom} f$ such that $\lim_{k \to \infty} h_y(x_k) = h_y^*$. Since $\operatorname{dom} f$ is bounded the sequence $(x_k)_{k \geq 1}$ has at least one limit point $\bar{x} \in \mathbb{R}^n$. Therefore, from the definition of a limit point, there exists a subsequence $(x_{k_l})_{l \geq 1} \subseteq (x_k)_{k \geq 1}$ such that $\lim_{l \to \infty} x_{k_l} = \bar{x}$. Since h_y is lower semicontinuous we have

$$h_y(\bar{x}) \leq \liminf_{l \to \infty} h_y(x_{k_l}) = \lim_{l \to \infty} h_y(x_{k_l}) = h_y^* = -\infty,$$

i. e. $h_y(\bar{x}) = -\infty$ which is a contradiction to the property of h_y being proper. Thus, $h_y^* > -\infty$. Since $h_y^* = h_y(\bar{x}) < \infty$ we have $\bar{x} \in \operatorname{dom} f$. In conclusion we have that for all $y \in \mathbb{R}^n$, $f^*(y) < \infty$ and therefore $\operatorname{dom} f^* = \mathbb{R}^n$. □

With the help of Lemma 4.2 we can now proof Theorem 4.1.

Proof. First, we can write the dual problem $(\mathrm{D}_{\mathrm{gen}})$ as

$$(\mathrm{D}_{\mathrm{gen}}) \qquad -\inf_{p\in\mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_m}} \{f^*(-K^*p)+g^*(p)\},$$

where $p=(p_1,\ldots,p_m)\in\mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_m}$, K^* and g^* are defined in the way like they have been introduced in Section 4.1. Now, assign the Fenchel dual problem

$$(\mathrm{D}_{\mathrm{D}_{\mathrm{gen}}}) \qquad -\sup_{y^*\in\mathbb{R}^n} \{-f^{**}(y^*)-g^{**}(K^{**}y^*)\}$$

to $(\mathrm{D}_{\mathrm{gen}})$ which can equivalently be written as

$$(\mathrm{D}_{\mathrm{D}_{\mathrm{gen}}}) \qquad \inf_{y^*\in\mathbb{R}^n} \{f^{**}(y^*)+g^{**}(Ky^*)\}.$$

Now, since f and g are proper, convex and lower semicontinuous functions, it holds that $f=f^{**}$ and $g=g^{**}$, respectively (see Theorem 2.8), and the dual problem $(\mathrm{D}_{\mathrm{D}_{\mathrm{gen}}})$ is nothing else than the original primal problem, i. e.

$$(\mathrm{D}_{\mathrm{D}_{\mathrm{gen}}}) \qquad \inf_{y^*\in\mathbb{R}^n} \{f(y^*)+g(Ky^*)\}.$$

Showing that strong duality between the primal-dual pair $(\mathrm{D}_{\mathrm{gen}})$ and $(\mathrm{D}_{\mathrm{D}_{\mathrm{gen}}})$ holds would yield the statements of the theorem and complete the proof. In general, for assuring strong duality, i. e. the case when the primal and the dual objective values coincide and the dual has an optimal solution a so-called qualification condition has to be fulfilled ([17, 12, 37, 77, 58]). For assuring strong duality between $(\mathrm{D}_{\mathrm{gen}})$ and $\left(\mathrm{D}_{\mathrm{D}_{\mathrm{gen}}}\right)$ we impose the interior-point type qualification condition (cf. [12, 58])

$$(QC^*) \qquad \mathrm{ri}(K^*(\mathrm{dom}\, g^*))\cap -\mathrm{ri}(\mathrm{dom}\, f^*)\neq\emptyset,$$

which can equivalently be written as

$$(QC^*) \qquad 0\in \mathrm{ri}\left(K^*(\mathrm{dom}\, g^*)+\mathrm{dom}\, f^*\right).$$

Thus, by [58, Corollary 31.2.1] we have that whenever (QC^*) is fulfilled it holds $v\left(\mathrm{D}_{\mathrm{gen}}\right)=v\left(\mathrm{D}_{\mathrm{D}_{\mathrm{gen}}}\right)$ and $\left(\mathrm{D}_{\mathrm{D}_{\mathrm{gen}}}\right)$ has an optimal solution, which are exactly the assertions of the theorem. Lets verify that (QC^*) is always fulfilled. From Lemma 4.2 we have that $\mathrm{dom}\, f^*=\mathbb{R}^n$ and $\mathrm{dom}\, g^*=\mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_m}$. Therefore $K^*(\mathrm{dom}\, g^*)+\mathrm{dom}\, f^*=\mathbb{R}^n$ and (QC^*) is always fulfilled and strong duality between $(\mathrm{D}_{\mathrm{gen}})$ and $\left(\mathrm{D}_{\mathrm{D}_{\mathrm{gen}}}\right)$ holds. In particular this means that $v\left(\mathrm{P}_{\mathrm{gen}}\right)=v\left(\mathrm{D}_{\mathrm{gen}}\right)$ and $(\mathrm{P}_{\mathrm{gen}})$ has an optimal solution. □

In addition we have, since we assumed $K(\mathrm{dom}\, f)\cap\mathrm{dom}\, g\neq\emptyset$, that $v\left(\mathrm{P}_{\mathrm{gen}}\right)=v\left(\mathrm{D}_{\mathrm{gen}}\right)\in\mathbb{R}$. Later we will assume that $(\mathrm{D}_{\mathrm{gen}})$ has an optimal solution, too, and that an upper bound on the norm of this solution is known.

4.3 Smoothing the general dual problem

In this section we will twice perform a regularization of the objective function of $(\mathrm{D}_{\mathrm{gen}})$ to obtain an optimization problem with a strongly convex and continuously differentiable objective function with Lipschitz-continuous gradient in order to apply a fast gradient method to solve the doubly smoothed dual and thus $(\mathrm{D}_{\mathrm{gen}})$ approximately. The first smoothing, aiming at making the objective function continuously differentiable with Lipschitz-continuous gradient, is performed in the next subsection. This regularization allows for solving $(\mathrm{P}_{\mathrm{gen}})$ approximately using an appropriate gradient method. We show later that the rate of convergence w. r. t. the optimal objective value is $O\left(\frac{1}{\varepsilon}\ln\left(\frac{1}{\varepsilon}\right)\right)$ with accuracy $\varepsilon > 0$. Since for first-order methods we have to require that the norm of the gradient is equal to zero at the optimal solution and we want to reconstruct an approximately feasible and optimal solution of the primal problem efficiently we have to solve the dual problem efficiently. Therefore, the second regularization making the objective strongly convex allows for the same rate of convergence of the norm of the gradient of the single smoothed function to zero referring to the same accuracy as obtained for the objective value up to a constant factor (see [36, 34, 35, 21]).

To conclude, performing twice a regularization serves for two purposes. First, we fulfill the assumptions on the objective function that allow for applying the fast gradient method presented in Subsection 2.3. Second, we are able to solve the dual problem and construct an approximate feasible and optimal primal solution efficiently.

4.3.1 First smoothing

In this subsection the first smoothing will be performed. Therefore, recall the dual problem

$$(\mathrm{D}_{\mathrm{gen}}) \qquad \sup_{(p_1,\ldots,p_m)\in\mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_m}} \left\{-f^*\left(-\sum_{i=1}^m K_i^* p_i\right) - \sum_{i=1}^m g_i^*\left(p_i\right)\right\}.$$

We introduce the function $F : \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m} \to \overline{\mathbb{R}}$ via

$$F(p_1,\ldots,p_m) := f^*\left(-\sum_{i=1}^m K_i^* p_i\right) + \sum_{i=1}^m g_i^*(p_i),$$

then the dual problem can equivalently be written as

$$(\mathrm{D}_{\mathrm{gen}}) \qquad -\inf_{(p_1,\ldots,p_m)\in\mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_m}} \left\{F(p_1,\ldots,p_m)\right\}.$$

In general, F is nondifferentiable since neither the function $(p_1,\ldots,p_m) \mapsto f^*\left(-\sum_{i=1}^m K_i^* p_i\right)$ nor the functions $p_i \mapsto g_i^*(p_i)$, $i = 1,\ldots,m$, can be guaranteed

to be smooth. Therefore, we now introduce smooth approximations to these functions. In the following we will use the euclidean norms in the different spaces $\mathbb{R}^{k_i}$, $i = 1, \ldots, m$, and $\mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$. We denote the euclidean norm in $\mathbb{R}^{k_i}$ by $\|\cdot\|_{k_i}$, $i = 1, \ldots, m$, and the norm in the product space by $\|\cdot\|_{\bar{k}}$, where this norm is defined by

$$\|(p_1, \ldots, p_m)\|_{\bar{k}} := \sqrt{\|p_1\|_{k_1}^2 + \ldots + \|p_m\|_{k_m}^2}$$

for $p = (p_1, \ldots, p_m)^{\mathrm{T}} \in \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$. The euclidean norm $\|\cdot\|$ with no subscript denotes the euclidean norm in $\mathbb{R}^n$. First, notice that

$$f^*\left(-\sum_{i=1}^m K_i^* p_i\right) = \sup_{x \in \operatorname{dom} f}\left\{\left\langle x, -\sum_{i=1}^m K_i^* p_i\right\rangle - f(x)\right\}. \tag{4.3}$$

This function will be approximated for a given real scalar $\rho > 0$ by $f_\rho^* : \mathbb{R}^n \to \mathbb{R}$ via

$$f_\rho^*\left(-\sum_{i=1}^m K_i^* p_i\right) := \sup_{x \in \operatorname{dom} f}\left\{\left\langle x, -\sum_{i=1}^m K_i^* p_i\right\rangle - f(x) - \frac{\rho}{2}\|x\|^2\right\}. \tag{4.4}$$

In a similar way, since

$$g_i^*(p_i) = \sup_{x \in \operatorname{dom} g_i}\left\{\langle x, p_i\rangle - g_i(x)\right\}, \tag{4.5}$$

this function will be approximated by $g_{i,\mu_i}^* : \mathbb{R}^{k_i} \to \mathbb{R}$ via

$$g_{i,\mu_i}^*(p_i) := \sup_{x \in \operatorname{dom} g_i}\left\{\langle x, p_i\rangle - g_i(x) - \frac{\mu_i}{2}\|x\|_{k_i}^2\right\} \tag{4.6}$$

for $i = 1, \ldots, m$ and a positive scalar $\mu_i > 0$, $i = 1, \ldots, m$. Notice that the objective functions in the maximization problems occurring in (4.4) and (4.6) are proper, strongly concave with parameter ρ and μ_i, $i = 1, \ldots, m$, respectively (cf. [40]), and upper semicontinuous and thus there always exist (see [9, Proposition A.8]) unique (see [9, Proposition B.10] optimal solutions to these problems. Using these approximations we define the function $F_{\rho,\mu} : \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m} \to \mathbb{R}$ by

$$F_{\rho,\mu}(p_1, \ldots, p_m) = f_\rho^*\left(-\sum_{i=1}^m K_i^* p_i\right) + \sum_{i=1}^m g_{i,\mu_i}^*(p_i) \tag{4.7}$$

which will be the objective of the smoothed dual problem

$$(\mathrm{D}_{\rho,\mu}) \qquad \inf_{(p_1,\ldots,p_m) \in \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}} \left\{F_{\rho,\mu}(p_1, \ldots, p_m)\right\}.$$

To see that $(\mathrm{D}_{\rho,\mu})$ has a continuously differentiable objective function with Lipschitz continuous gradient we state the following proposition.

Proposition 4.3. *(a) The functions $g^*_{i,\mu_i} : \mathbb{R}^{k_i} \to \mathbb{R}$ defined by (4.6) are continuously differentiable for all $i = 1, \ldots, m$.*
(b) The function $f^*_\rho : \mathbb{R}^n \to \mathbb{R}$ defined by (4.4) is continuously differentiable.

Proof. (a) For all $i = 1, \ldots, m$ we reformulate the function g^*_{i,μ_i} as follows.

$$\begin{aligned}
-g^*_{i,\mu_i}(p_i) &= -\sup_{x\in\mathbb{R}^{k_i}} \left\{ \langle x, p_i\rangle - g_i(x) - \frac{\mu_i}{2}\|x\|^2_{k_i} \right\} \\
&= \inf_{x\in\mathbb{R}^{k_i}} \left\{ g_i(x) + \frac{\mu_i}{2}\|x\|^2_{k_i} - \langle x, p_i\rangle \right\} \\
&= \inf_{x\in\mathbb{R}^{k_i}} \left\{ g_i(x) + \frac{\mu_i}{2}\langle x, x\rangle - 2\frac{1}{2}\langle x, p_i\rangle + \frac{1}{2\mu_i}\langle p_i, p_i\rangle - \frac{1}{2\mu_i}\langle p_i, p_i\rangle \right\} \\
&= \inf_{x\in\mathbb{R}^{k_i}} \left\{ g_i(x) + \frac{\mu_i}{2}\langle x, x\rangle - 2\frac{1}{2}\langle x, p_i\rangle + \frac{1}{2\mu_i}\langle p_i, p_i\rangle \right\} - \frac{1}{2\mu_i}\langle p_i, p_i\rangle \\
&= \inf_{x\in\mathbb{R}^{k_i}} \left\{ g_i(x) + \frac{\mu_i}{2}\left(\langle x, x\rangle - 2\left\langle x, \frac{p_i}{\mu_i}\right\rangle + \left\langle \frac{p_i}{\mu_i}, \frac{p_i}{\mu_i}\right\rangle \right) \right\} - \frac{1}{2\mu_i}\|p_i\|^2_{k_i} \\
&= \inf_{x\in\mathbb{R}^{k_i}} \left\{ g_i(x) + \frac{\mu_i}{2}\left\| \frac{p_i}{\mu_i} - x \right\|^2_{k_i} \right\} - \frac{1}{2\mu_i}\|p_i\|^2_{k_i}.
\end{aligned}$$

Analyzing the minimization problem in the last line of the above calculations we observe that this is the Moreau envelop (cf. Section 2.1) of the function g_i of parameter $\frac{1}{\mu_i}$ at the point $\frac{p_i}{\mu_i}$, i. e.

$$-g^*_{i,\mu_i}(p_i) = {}^{\frac{1}{\mu_i}}g_i\left(\frac{p_i}{\mu_i}\right) - \frac{1}{2\mu_i}\|p_i\|^2_{k_i}, \tag{4.8}$$

for all $i = 1, \ldots, m$. From the observations in Section 2.1 we get that ${}^{\frac{1}{\mu_i}}g_i$ is differentiable and so g^*_{i,μ_i} is.
(b) Applying the same reasoning for the function f^*_ρ we get

$$\begin{aligned}
-f^*_\rho\left(-\sum_{i=1}^m K_i^* p_i\right) &= -\sup_{x\in\mathbb{R}^n} \left\{ \left\langle x, -\sum_{i=1}^m K_i^* p_i \right\rangle - f(x) - \frac{\rho}{2}\|x\|^2 \right\} \\
&= \inf_{x\in\mathbb{R}^n} \left\{ f(x) + \frac{\rho}{2}\left\| \frac{-\sum_{i=1}^m K_i^* p_i}{\rho} - x \right\|^2 \right\} - \frac{1}{2\rho}\left\| \sum_{i=1}^m K_i^* p_i \right\|^2 .
\end{aligned}$$

Again, in the last line we recognize the Moreau envelop of f of parameter $\frac{1}{\rho}$ at point $-\frac{1}{\rho}\sum_{i=1}^m K_i^* p_i$, i. e.

$$-f_\rho^*\left(-\sum_{i=1}^m K_i^* p_i\right) = {}^{\frac{1}{\rho}}f\left(\frac{-\sum_{i=1}^m K_i^* p_i}{\rho}\right) - \frac{1}{2\rho}\left\|\sum_{i=1}^m K_i^* p_i\right\|^2 \tag{4.9}$$

Since ${}^{\frac{1}{\rho}}f$ is differentiable with Lipschitz continuous gradient the same holds for f_ρ^*. □

The following theorem states important properties of $F_{\rho,\mu}$ and its gradient as well as its Lipschitz constant.

Theorem 4.4. *The function $F_{\rho,\mu} : \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m} \to \mathbb{R}$ defined by (4.7) is continuously differentiable with Lipschitz continuous gradient*

$$\nabla F_{\rho,\mu}(p_1,\ldots,p_m) = \left(\mathrm{Prox}_{\frac{1}{\mu_1}g_1}\left(\frac{p_1}{\mu_1}\right),\ldots,\mathrm{Prox}_{\frac{1}{\mu_m}g_m}\left(\frac{p_m}{\mu_m}\right)\right)$$

$$-\left(K_1\,\mathrm{Prox}_{\frac{1}{\rho}f}\left(-\frac{\sum_{i=1}^m K_i^* p_i}{\rho}\right),\ldots,K_m\,\mathrm{Prox}_{\frac{1}{\rho}f}\left(-\frac{\sum_{i=1}^m K_i^* p_i}{\rho}\right)\right). \tag{4.10}$$

Moreover, the Lipschitz constant of the gradient $\nabla F_{\rho,\mu}(p_1,\ldots,p_m)$ is given by

$$L(\rho,\mu) = \max_{i=1,\ldots,m}\left\{\frac{1}{\mu_i}\right\} + \frac{\sqrt{m}}{\rho}\left[\left(\sum_{i=1}^m \|K_i\|^2\right)\max_{i=1,\ldots,m}\{\|K_i\|^2\}\right]^{\frac{1}{2}}. \tag{4.11}$$

Proof. The continuous differentiability of $F_{\rho,\mu}$ is a direct consequence of Proposition 4.3 as it is the sum of continuously differentiable functions. Next we calculate the gradient of $F_{\rho,\mu}$ in order to obtain its Lipschitz constant. We first recall that $K^* : \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m} \to \mathbb{R}^n$, $K^*(p_1,\ldots,p_m) = \sum_{i=1}^m K_i^* p_i$, is the adjoint operator of the linear operator $K : \mathbb{R}^n \to \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$, $Kx = (K_1 x,\ldots,K_m x)$ (cf. Section 4.1) and therefore we get

$$\nabla F_{\rho,\mu}(p_1,\ldots,p_m) = \nabla\left(f_\rho^* \circ (-K^*)\right)(p_1,\ldots,p_m) + \left(\nabla g_{1,\mu_1}^*(p_1),\ldots,\nabla g_{m,\mu_m}^*(p_m)\right)$$

for the gradient of $F_{\rho,\mu}$. We derive the first summand of the above formula for which it holds by the chain rule

$$\nabla(f_\rho^* \circ (-K^*))(p_1,\ldots,p_m) = (-K^*)^* \nabla f_\rho^*(-K^*(p_1,\ldots,p_m)).$$

Further, we have for all $z \in \mathbb{R}^n$ by taking into account (4.9) and (2.6),

$$\nabla f_\rho^*(z) = -\nabla {}^{\frac{1}{\rho}}f\left(\frac{z}{\rho}\right) + \nabla\left(\frac{1}{2\rho}\|\cdot\|^2\right)(z)$$

$$= -\frac{1}{\rho}\left(\rho\left(\frac{z}{\rho} - \operatorname{Prox}_{\frac{1}{\rho}f}\left(\frac{z}{\rho}\right)\right)\right) + \frac{z}{\rho}$$
$$= \operatorname{Prox}_{\frac{1}{\rho}f}\left(\frac{z}{\rho}\right),$$

i. e.

$$\nabla f_\rho^*\left(-K^*(p_1,\ldots,p_m)\right) = \operatorname{Prox}_{\frac{1}{\rho}f}\left(-\frac{\sum_{i=1}^m K_i^* p_i}{\rho}\right),$$

and, finally, putting all together,

$$\nabla\left(f_\rho^* \circ (-K^*)\right)(p_1,\ldots,p_m) = -K\left(\operatorname{Prox}_{\frac{1}{\rho}f}\left(-\frac{\sum_{i=1}^m K_i^* p_i}{\rho}\right)\right)$$
$$= \left(-K_1 \operatorname{Prox}_{\frac{1}{\rho}f}\left(-\frac{\sum_{i=1}^m K_i^* p_i}{\rho}\right),\ldots,-K_m \operatorname{Prox}_{\frac{1}{\rho}f}\left(-\frac{\sum_{i=1}^m K_i^* p_i}{\rho}\right)\right).$$

For the second summand in the formula for $\nabla F_{\rho,\mu}$ we have by taking into account (4.8) and (2.6)

$$\nabla g_{i,\mu_i}^*(p_i) = -\nabla^{\frac{1}{\mu_i}} g_i\left(\frac{p_i}{\mu_i}\right) + \nabla\left(\frac{1}{2\mu_i}\|\cdot\|_{k_i}^2\right)(p_i)$$
$$= -\frac{1}{\mu_i}\left(\mu_i\left(\frac{p_i}{\mu_i} - \operatorname{Prox}_{\frac{1}{\mu_i}g_i}\left(\frac{p_i}{\mu_i}\right)\right)\right) + \frac{p_i}{\mu_i}$$
$$= \operatorname{Prox}_{\frac{1}{\mu_i}g_i}\left(\frac{p_i}{\mu_i}\right)$$

for all $i = 1,\ldots,m$. Thus, the gradient $\nabla F_{\rho,\mu}$ becomes

$$\nabla F_{\rho,\mu}(p_1,\ldots,p_m) = \left(-K_1 \operatorname{Prox}_{\frac{1}{\rho}f}\left(-\frac{\sum_{i=1}^m K_i^* p_i}{\rho}\right),\ldots,\right.$$
$$\left.-K_m \operatorname{Prox}_{\frac{1}{\rho}f}\left(-\frac{\sum_{i=1}^m K_i^* p_i}{\rho}\right)\right)$$
$$+\left(\operatorname{Prox}_{\frac{1}{\mu_1}g_1}\left(\frac{p_1}{\mu_1}\right),\ldots,\operatorname{Prox}_{\frac{1}{\mu_m}g_m}\left(\frac{p_m}{\mu_m}\right)\right). \qquad (4.12)$$

In order to obtain the Lipschitz constant $L(\rho,\mu) := L(\rho) + L(\mu)$ of the gradient $\nabla F_{\rho,\mu}$, where $L(\rho)$ and $L(\mu)$ denote the Lipschitz constants of the first and second summand occurring in the gradient formula, respectively, we calculate

both components separately. Therefore, notice that from [38, Lemma 6.36] we get that the proximal mapping is Lipschitz continuous with Lipschitz constant equal to 1. We first observe the Lipschitz constant $L(\mu)$ of the second summand in (4.10). Therefore, let $(p_1, \ldots, p_m), (p_1', \ldots, p_m') \in \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$ and obtain

$$\begin{aligned}
&\left\| \left(\operatorname{Prox}_{\frac{1}{\mu_1} g_1} \left(\frac{p_1}{\mu_1} \right), \ldots, \operatorname{Prox}_{\frac{1}{\mu_m} g_m} \left(\frac{p_m}{\mu_m} \right) \right) - \right. \\
&\qquad \left. \left(\operatorname{Prox}_{\frac{1}{\mu_1} g_1} \left(\frac{p_1'}{\mu_1} \right), \ldots, \operatorname{Prox}_{\frac{1}{\mu_m} g_m} \left(\frac{p_m'}{\mu_m} \right) \right) \right\|_{\bar{k}}^2 \\
&= \left\| \left(\operatorname{Prox}_{\frac{1}{\mu_1} g_1} \left(\frac{p_1}{\mu_1} \right) - \operatorname{Prox}_{\frac{1}{\mu_1} g_1} \left(\frac{p_1'}{\mu_1} \right), \right. \right. \\
&\qquad \left. \left. \ldots, \operatorname{Prox}_{\frac{1}{\mu_m} g_m} \left(\frac{p_m}{\mu_m} \right) - \operatorname{Prox}_{\frac{1}{\mu_m} g_m} \left(\frac{p_m'}{\mu_m} \right) \right) \right\|_{\bar{k}}^2 \\
&= \sum_{i=1}^{m} \left\| \operatorname{Prox}_{\frac{1}{\mu_i} g_i} \left(\frac{p_i}{\mu_i} \right) - \operatorname{Prox}_{\frac{1}{\mu_i} g_i} \left(\frac{p_i'}{\mu_i} \right) \right\|_{k_i}^2 \\
&\leq \sum_{i=1}^{m} \left\| \frac{p_i}{\mu_i} - \frac{p_i'}{\mu_i} \right\|_{k_i}^2 = \sum_{i=1}^{m} \frac{1}{\mu_i^2} \left\| p_i - p_i' \right\|_{k_i}^2 \\
&\leq \max_{i=1,\ldots,m} \left\{ \frac{1}{\mu_i^2} \right\} \sum_{i=1}^{m} \left\| p_i - p_i' \right\|_{k_i}^2 = \max_{i=1,\ldots,m} \left\{ \frac{1}{\mu_i^2} \right\} \left\| (p_1 - p_1', \ldots, p_m - p_m') \right\|_{\bar{k}}^2 \\
&= \max_{i=1,\ldots,m} \left\{ \frac{1}{\mu_i^2} \right\} \left\| (p_1, \ldots, p_m) - (p_1', \ldots, p_m') \right\|_{\bar{k}}^2,
\end{aligned}$$

i. e. we have

$$L(\mu) = \left[\max_{i=1,\ldots,m} \left\{ \frac{1}{\mu_i^2} \right\} \right]^{\frac{1}{2}} = \max_{i=1,\ldots,m} \left\{ \frac{1}{\mu_i} \right\}.$$

To get the Lipschitz constant $L(\rho)$ of the first summand and hence the Lipschitz constant $L(\rho, \mu)$ for the gradient of $F_{\rho,\mu}$ we calculate again for arbitrary $(p_1, \ldots, p_m), (p_1', \ldots, p_m') \in \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$,

$$\begin{aligned}
&\left\| \left(-K_1 \operatorname{Prox}_{\frac{1}{\rho} f} \left(-\frac{\sum_{i=1}^{m} K_i^* p_i}{\rho} \right), \ldots, -K_m \operatorname{Prox}_{\frac{1}{\rho} f} \left(-\frac{\sum_{i=1}^{m} K_i^* p_i}{\rho} \right) \right) - \right. \\
&\qquad \left. \left(-K_1 \operatorname{Prox}_{\frac{1}{\rho} f} \left(-\frac{\sum_{i=1}^{m} K_i^* p_i'}{\rho} \right), \ldots, -K_m \operatorname{Prox}_{\frac{1}{\rho} f} \left(-\frac{\sum_{i=1}^{m} K_i^* p_i'}{\rho} \right) \right) \right\|_{\bar{k}}^2
\end{aligned}$$

$$= \sum_{j=1}^{m} \left\| -K_j \left(\mathrm{Prox}_{\frac{1}{\rho} f} \left(-\frac{\sum_{i=1}^{m} K_i^* p_i}{\rho} \right) - \mathrm{Prox}_{\frac{1}{\rho} f} \left(-\frac{\sum_{i=1}^{m} K_i^* p_i'}{\rho} \right) \right) \right\|_{k_j}^2$$

$$\leq \left(\sum_{j=1}^{m} \|K_j\|^2 \right) \left\| \left(\mathrm{Prox}_{\frac{1}{\rho} f} \left(-\frac{\sum_{i=1}^{m} K_i^* p_i}{\rho} \right) - \mathrm{Prox}_{\frac{1}{\rho} f} \left(-\frac{\sum_{i=1}^{m} K_i^* p_i'}{\rho} \right) \right) \right\|^2$$

and by using the property of the proximal mapping being nonexpansive we continue

$$\leq \left(\sum_{j=1}^{m} \|K_j\|^2 \right) \left\| \frac{1}{\rho} \left(-\sum_{i=1}^{m} K_i^* p_i \right) - \frac{1}{\rho} \left(-\sum_{i=1}^{m} K_i^* p_i' \right) \right\|^2$$

$$= \frac{1}{\rho^2} \left(\sum_{j=1}^{m} \|K_j\|^2 \right) \left\| \sum_{i=1}^{m} K_i^* p_i - \sum_{i=1}^{m} K_i^* p_i' \right\|^2$$

$$= \frac{1}{\rho^2} \left(\sum_{j=1}^{m} \|K_j\|^2 \right) \|K_1^*(p_1 - p_1') + \ldots + K_m^*(p_m - p_m')\|^2$$

$$\leq \frac{m}{\rho^2} \left(\sum_{j=1}^{m} \|K_j\|^2 \right) \sum_{j=1}^{m} \|K_j^*(p_j - p_j')\|^2$$

$$\leq \frac{m}{\rho^2} \left(\sum_{j=1}^{m} \|K_j\|^2 \right) \sum_{j=1}^{m} \|K_j^*\|^2 \|(p_j - p_j')\|_{k_j}^2$$

$$\leq \frac{m}{\rho^2} \left(\sum_{j=1}^{m} \|K_j\|^2 \right) \max_{j=1,\ldots,m} \{\|K_j\|^2\} \sum_{j=1}^{m} \|p_j - p_j'\|_{k_j}^2$$

$$= \frac{m}{\rho^2} \left(\sum_{j=1}^{m} \|K_j\|^2 \right) \max_{j=1,\ldots,m} \{\|K_j\|^2\} \|(p_1, \ldots, p_m) - (p_1', \ldots, p_m')\|_{\bar{k}}^2.$$

Thus we obtain the Lipschitz constant for the first summand in (4.10)

$$L(\rho) = \frac{\sqrt{m}}{\rho} \left[\left(\sum_{i=1}^{m} \|K_i\|^2 \right) \max_{i=1,\ldots,m} \{\|K_i\|^2\} \right]^{\frac{1}{2}}$$

and therefore, the gradient of $F_{\rho,\mu}$ is Lipschitz continuous with Lipschitz constant $L(\rho, \mu) = L(\rho) + L(\mu)$ which is (4.11) stated in the theorem. □

Until now we have regularized the objective function of $(\mathrm{D}_{\mathrm{gen}})$ and arrived at $F_{\rho,\mu}$, the objective function of $(\mathrm{D}_{\rho,\mu})$, which is now continuously differentiable with Lipschitz continuous gradient with constant $L(\rho,\mu)$. As mentioned above, this single regularization is not sufficient to construct an approximate solution to the primal problem $(\mathrm{P}_{\mathrm{gen}})$ efficiently which actually is our goal.

4.3.2 Second smoothing

In order to get an optimization problem with strongly convex objective function we apply a regularization to $F_{\rho,\mu}$, i. e. we introduce for $\gamma > 0$ the function $F_{\rho,\mu,\gamma} : \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m} \to \mathbb{R}$,

$$F_{\rho,\mu,\gamma}(p_1,\ldots,p_m) = F_{\rho,\mu}(p_1,\ldots,p_m) + \frac{\gamma}{2}\|(p_1,\ldots,p_m)\|_{\bar{k}}^2, \tag{4.13}$$

which is strongly convex with modulus γ (see [40, Proposition B 1.1.2]) and will be the objective function of the optimization problem

$$(\mathrm{D}_{\rho,\mu,\gamma}) \qquad \inf_{(p_1,\ldots,p_m)\in\mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_m}} \{F_{\rho,\mu,\gamma}(p_1,\ldots,p_m)\}.$$

The next theorem establishes further properties of $F_{\rho,\mu,\gamma}$.

Theorem 4.5. *The function $F_{\rho,\mu,\gamma} : \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m} \to \mathbb{R}$ given by (4.13) is continuously differentiable with Lipschitz continuous gradient. Moreover, the Lipschitz constant of $\nabla F_{\rho,\mu,\gamma}(p_1,\ldots,p_m)$ is given by*

$$L(\rho,\mu,\gamma) = \max_{i=1,\ldots,m}\left\{\frac{1}{\mu_i}\right\} + \frac{\sqrt{m}}{\rho}\left[\left(\sum_{i=1}^{m}\|K_i\|^2\right)\max_{i=1,\ldots,m}\{\|K_i\|^2\}\right]^{\frac{1}{2}} + \gamma. \tag{4.14}$$

Proof. Theorem 4.4 together with the fact that the last term in (4.13) is continuously differentiable yields the continuous differentiability of $F_{\rho,\mu,\gamma}$. The gradient of the new objective $F_{\rho,\mu,\gamma}$ is obviously Lipschitz continuous and is given by

$$\nabla F_{\rho,\mu,\gamma}(p_1,\ldots,p_m) = \nabla F_{\rho,\mu}(p_1,\ldots,p_m) + \gamma(p_1,\ldots,p_m), \tag{4.15}$$

where the Lipschitz constant $L(\rho,\mu,\gamma)$ of $\nabla F_{\rho,\mu,\gamma}$ has yet to be calculated. For arbitrary $(p_1,\ldots,p_m),(p_1',\ldots,p_m') \in \mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_m}$ consider

$$\begin{aligned}
&\|\nabla F_{\rho,\mu,\gamma}(p_1,\ldots,p_m) - \nabla F_{\rho,\mu,\gamma}(p_1',\ldots,p_m')\|_{\bar{k}} \\
&\quad\le \|\nabla F_{\rho,\mu}(p_1,\ldots,p_m) - \nabla F_{\rho,\mu}(p_1',\ldots,p_m')\|_{\bar{k}} + \gamma\|(p_1,\ldots,p_m)-(p_1',\ldots,p_m')\|_{\bar{k}} \\
&\quad\le (L(\rho,\mu)+\gamma)\|(p_1,\ldots,p_m)-(p_1',\ldots,p_m')\|_{\bar{k}},
\end{aligned}$$

i. e. the Lipschitz constant of the gradient of $F_{\rho,\mu,\gamma}$ results in (4.14). □

The objective function $F_{\rho,\mu,\gamma}$ now matches the conditions required to apply the fast gradient method (see section 2.3) introduced by Nesterov in [50] to solve problem $(\mathrm{D}_{\rho,\mu,\gamma})$ efficiently. In the next subsection the algorithmic scheme is presented and some convergence properties are derived in order to obtain a solution to $(\mathrm{D}_{\rho,\mu,\gamma})$ and to ensure the ability of constructing an approximate solution to $(\mathrm{P}_{\mathrm{gen}})$ with the help of the iterates generated by the fast gradient algorithm.

4.4 Applying the fast gradient method

In this section we apply the fast gradient method to the doubly smoothed dual problem $(\mathrm{D}_{\rho,\mu,\gamma})$. In the following we will denote by $p^k = (p_1^k, \ldots, p_m^k) \in \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$ the $k-$th iterate produced by the fast gradient scheme presented in the box below (Figure 4.1). Further, we will assume that there exists an optimal solution to $(\mathrm{D}_{\mathrm{gen}})$, denoted by $p^* = (p_1^*, \ldots, p_m^*) \in \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$, while $F^* = F(p^*)$ will denote the corresponding optimal objective value. In addition we require that there exists an upper bound $R > 0$ for the norm of p^*, i. e. $\|p^*\|_{\bar{k}} \leq R$, which is assumed to be known. Finally, let $\bar{p}^* = (\bar{p}_1^*, \ldots, \bar{p}_m^*) \in \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$ and $F_{\rho,\mu,\gamma}^* = F_{\rho,\mu,\gamma}(\bar{p}^*)$ denote the optimal solution to $(\mathrm{D}_{\rho,\mu,\gamma})$ and the corresponding optimal objective value, respectively.

Fast Gradient Method

Initialization: set $w^0 = p^0 = 0 \in \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$

Iteration $k \geq 0$: set

$$p^{k+1} = w^k - \frac{1}{L(\rho,\mu,\gamma)} \nabla F_{\rho,\mu,\gamma}\left(w^k\right)$$

and

$$w^{k+1} = p^{k+1} + \frac{\sqrt{L(\rho,\mu,\gamma)} - \sqrt{\gamma}}{\sqrt{L(\rho,\mu,\gamma)} + \sqrt{\gamma}} \left[p^{k+1} - p^k\right]$$

Figure 4.1: Fast gradient scheme.

In the following subsections we will investigate convergence properties that can be guaranteed for the sequence $(p^k)_{k\geq 0}$ generated by the fast gradient scheme which is presented here again in terms of the variables used in this chapter for the sake of clarity.

4.4.1 Convergence of the optimal objective value

The main issue of this subsection is to verify that for the sequence $(p^k)_{k\geq 0} \subseteq \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$ it can be ensured that $F(p^k)$ converges to the optimal objective value $v(\mathrm{D}_{\mathrm{gen}})$. It is worth noticing that the iterates p^k, $k \geq 0$, are in fact iterates of the dual variables of the modified problem $(\mathrm{D}_{\rho,\mu,\gamma})$. Nevertheless, the convergence we will show in this subsection holds for the values of the objective function of $(\mathrm{D}_{\mathrm{gen}})$, the optimization problem we want to solve approximately.

Let us first introduce some important constants that will allow for an estimate in the following. Let be

$$D_{g_i} := \sup_{x \in \operatorname{dom} g_i} \left\{ \frac{1}{2}\|x\|^2 \right\}, \forall i = 1, \ldots, m, \quad \text{and} \quad D_f := \sup_{x \in \operatorname{dom} f} \left\{ \frac{1}{2}\|x\|^2 \right\} \tag{4.16}$$

and notice that $D_{g_i} \in \mathbb{R}$ for all $i = 1, \ldots, m$ and $D_f \in \mathbb{R}$ since the effective domains of the functions f and g_i were assumed to be bounded.

Proposition 4.6. *Let f_ρ^* and g_{i,μ_i}^*, $i = 1, \ldots, m$, be given by (4.4) and (4.6) and $\rho > 0$ and $\mu_i > 0$, $i = 1, \ldots, m$ the corresponding smoothing parameters. Further, let $K_i^* : \mathbb{R}^{k_i} \to \mathbb{R}^n$, $i = 1, \ldots, m$, be the adjoint operators of the operators K_i, $i = 1, \ldots, m$, in (P_{gen}) and D_f and D_{g_i} the constants (4.16). Then, for all $p = (p_1, \ldots, p_m) \in \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$ it holds*

$$(i) \quad f_\rho^*\left(-\sum_{i=1}^m K_i^* p_i\right) \leq f^*\left(-\sum_{i=1}^m K_i^* p_i\right) \leq f_\rho^*\left(-\sum_{i=1}^m K_i^* p_i\right) + \rho D_f$$

and

$$(ii) \quad g_{i,\mu_i}^*(p_i) \leq g_i^*(p_i) \leq g_{i,\mu_i}^*(p_i) + \mu_i D_{g_i} \qquad \forall i = 1, \ldots, m.$$

Proof. We show the relations in (i). One has

$$\begin{aligned} f_\rho^*\left(-\sum_{i=1}^m K_i^* p_i\right) &= \sup_{x \in \operatorname{dom} f} \left\{ \left\langle x, -\sum_{i=1}^m K_i^* p_i \right\rangle - f(x) - \frac{\rho}{2}\|x\|^2 \right\} \\ &\leq \sup_{x \in \operatorname{dom} f} \left\{ \left\langle x, -\sum_{i=1}^m K_i^* p_i \right\rangle - f(x) \right\} = f^*\left(-\sum_{i=1}^m K_i^* p_i\right) \\ &= \sup_{x \in \operatorname{dom} f} \left\{ \left\langle x, -\sum_{i=1}^m K_i^* p_i \right\rangle - f(x) - \frac{\rho}{2}\|x\|^2 + \frac{\rho}{2}\|x\|^2 \right\} \\ &\leq \sup_{x \in \operatorname{dom} f} \left\{ \left\langle x, -\sum_{i=1}^m K_i^* p_i \right\rangle - f(x) - \frac{\rho}{2}\|x\|^2 \right\} \end{aligned}$$

$$+ \sup_{x \in \operatorname{dom} f} \left\{ \frac{\rho}{2} \|x\|^2 \right\}$$

$$= f_\rho^* \left(-\sum_{i=1}^m K_i^* p_i \right) + \rho D_f.$$

The proof of the relations in (ii) follows similarly. □

An immediate consequence of Proposition 4.6 is the following corollary.

Corollary 4.7. *Let be $F : \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m} \to \overline{\mathbb{R}}$ the objective function of problem (D_{gen}) and $F_{\rho,\mu} : \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m} \to \mathbb{R}$ the objective function of the corresponding dual problem $(D_{\rho,\mu})$. Further, let D_f and D_{g_i}, $i = 1, \ldots, m$, be the constants defined by (4.16) and $\rho > 0$ and $\mu_i > 0$, $i = 1, \ldots, m$, the smoothing parameters w. r. t. the functions f_ρ^* and g_{i,μ_i}^*, respectively. Then for all $p = (p_1, \ldots, p_m) \in \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$ it holds*

$$F_{\rho,\mu}(p) \leq F(p) \leq F_{\rho,\mu}(p) + \rho D_f + \sum_{i=1}^m \mu_i D_{g_i}. \tag{4.17}$$

Proof. Summing up the inequalities (i) and (ii) from Proposition 4.6 yields the assertion of the corollary. □

The next result establishes an upper bound on the distance between the objective values of (D_{gen}) at each iterate $p^k = (p_1^k, \ldots, p_m^k) \in \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$ and its optimal objective value at the optimal solution $p^* = (p_1^*, \ldots, p_m^*) \in \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$ assumed to exist.

Proposition 4.8. *Let $(p^k)_{k \geq} 0$ be the sequence of iterates generated by the algorithmic scheme 4.1 and $F : \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m} \to \overline{\mathbb{R}}$ be the objective function of the problem (D_{gen}). Further, let p^* be the optimal solution to it, $R > 0$ such that $\|p^*\|_{\bar{k}} \leq R$ and ρ, μ_i, D_f, D_{g_i}, $i = 1, \ldots, m$, like in Proposition 4.6 and $\gamma > 0$ the smoothing parameter in (4.13). Then it holds*

$$F(p^k) - F(p^*) \leq (2 + \sqrt{2}) \left(F(0) - F(p^*) + \rho D_f + \sum_{i=1}^m \mu_i D_{g_i} \right) \mathrm{e}^{-\frac{1}{2} k \sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}}$$
$$+ \rho D_f + \sum_{i=1}^m \mu_i D_{g_i} + \frac{\gamma}{2} R^2. \tag{4.18}$$

for all $k \geq 0$.

Proof. First, since $p^0 = 0$ and from the definition of $F_{\rho,\mu,\gamma}$ (cf. (4.13)) we have

$$F_{\rho,\mu,\gamma}(0) = F_{\rho,\mu}(0) = f_\rho^*(0) + \sum_{i=1}^m g_{i,\mu_i}^*(0) \tag{4.19}$$

and

$$F_{\rho,\mu,\gamma}(\bar{p}^*) = F_{\rho,\mu}(\bar{p}^*) + \frac{\gamma}{2}\|\bar{p}^*\|_{\bar{k}}^2. \tag{4.20}$$

From (2.19) we get the inequality

$$\frac{\gamma}{2}\left\|p^k - \bar{p}^*\right\|_{\bar{k}}^2 \leq F_{\rho,\mu,\gamma}(p^k) - F^*_{\rho,\mu,\gamma} \tag{4.21}$$

which means for $k = 0$, i. e. $p^0 = 0$,

$$\frac{\gamma}{2}\|p^0 - \bar{p}^*\|_{\bar{k}}^2 = \frac{\gamma}{2}\|\bar{p}^*\|_{\bar{k}}^2 \leq F_{\rho,\mu,\gamma}(0) - F^*_{\rho,\mu,\gamma} = F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*) - \frac{\gamma}{2}\|\bar{p}^*\|_{\bar{k}}^2, \tag{4.22}$$

or, equivalently,

$$\|\bar{p}^*\|_{\bar{k}}^2 \leq \frac{1}{\gamma}(F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*)). \tag{4.23}$$

From (2.20) we get the relation

$$F_{\rho,\mu,\gamma}(p^k) - F^*_{\rho,\mu,\gamma} \leq \left(F_{\rho,\mu,\gamma}(0) - F^*_{\rho,\mu,\gamma} + \frac{\gamma}{2}\|p^0 - \bar{p}^*\|_{\bar{k}}^2\right) \mathrm{e}^{-k\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}}, \tag{4.24}$$

for arbitrary $k \geq 0$ and thus, for $k > 0$, by taking into consideration (4.21) we get

$$\begin{aligned}
\left\|p^k - \bar{p}^*\right\|_{\bar{k}}^2 &\leq \frac{2}{\gamma}\left(F_{\rho,\mu,\gamma}(p^k) - F_{\rho,\mu,\gamma}(\bar{p}^*)\right) \\
&\leq \frac{2}{\gamma}\left(F_{\rho,\mu,\gamma}(0) - F^*_{\rho,\mu,\gamma} + \frac{\gamma}{2}\|\bar{p}^*\|_{\bar{k}}^2\right) \mathrm{e}^{-k\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} \\
&= \frac{2}{\gamma}\left(F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*)\right) \mathrm{e}^{-k\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}}
\end{aligned} \tag{4.25}$$

by accounting for the definition of $F_{\rho,\mu,\gamma}$. For the distance of the values of $F_{\rho,\mu}$ at points p^k and $\bar{p}^*$ we get (cf. (4.24))

$$\begin{aligned}
F_{\rho,\mu}(p^k) - F_{\rho,\mu}(\bar{p}^*) &= F_{\rho,\mu,\gamma}(p^k) - \frac{\gamma}{2}\|p^k\|_{\bar{k}}^2 - F_{\rho,\mu,\gamma}(\bar{p}^*) + \frac{\gamma}{2}\|\bar{p}^*\|_{\bar{k}}^2 \\
&= F_{\rho,\mu,\gamma}(p^k) - F_{\rho,\mu,\gamma}(\bar{p}^*) + \frac{\gamma}{2}\left(\|\bar{p}^*\|_{\bar{k}}^2 - \left\|p^k\right\|_{\bar{k}}^2\right) \\
&\leq \left(F_{\rho,\mu,\gamma}(0) - F^*_{\rho,\mu,\gamma} + \frac{\gamma}{2}\|\bar{p}^*\|_{\bar{k}}^2\right) \mathrm{e}^{-k\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} + \frac{\gamma}{2}\left(\|\bar{p}^*\|_{\bar{k}}^2 - \left\|p^k\right\|_{\bar{k}}^2\right) \\
&= \left(F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*)\right) \mathrm{e}^{-k\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} + \frac{\gamma}{2}\left(\|\bar{p}^*\|_{\bar{k}}^2 - \left\|p^k\right\|_{\bar{k}}^2\right).
\end{aligned} \tag{4.26}$$

Lets have a closer look at the last summand of (4.26). Notice that by the reverse triangle inequality

$$\left| \|\bar{p}^*\|_{\bar{k}} - \left\|p^k\right\|_{\bar{k}} \right| \leq \left\|\bar{p}^* - p^k\right\|_{\bar{k}},$$

and it holds the estimate

$$\left\|p^k\right\|_{\bar{k}} = \left\|p^k - \bar{p}^* + \bar{p}^*\right\|_{\bar{k}} \leq \left\|p^k - \bar{p}^*\right\|_{\bar{k}} + \|\bar{p}^*\|_{\bar{k}}.$$

Further, since $F_{\rho,\mu,\gamma}$ is strongly convex with parameter γ we get from (2.18) for all $k \geq 0$

$$F_{\rho,\mu,\gamma}(p^k) \geq F_{\rho,\mu,\gamma}(\bar{p}^*) + \frac{\gamma}{2}\left\|p^k - \bar{p}^*\right\|_{\bar{k}}^2,$$

i. e., since $p^0 = 0$ (cf. (4.22)),

$$\begin{aligned} \|\bar{p}^*\|_{\bar{k}}^2 &\leq \frac{2}{\gamma}\left(F_{\rho,\mu,\gamma}(0) - F_{\rho,\mu,\gamma}(\bar{p}^*)\right) \\ &= \frac{2}{\gamma}\left(F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*) - \frac{\gamma}{2}\|\bar{p}^*\|_{\bar{k}}^2\right). \end{aligned} \tag{4.27}$$

Thus, we get the estimate

$$\|\bar{p}^*\|_{\bar{k}} \leq \frac{1}{\sqrt{\gamma}}\sqrt{F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*)}. \tag{4.28}$$

Investigating the last term in brackets in (4.26) we get

$$\begin{aligned} \|\bar{p}^*\|_{\bar{k}}^2 - \left\|p^k\right\|_{\bar{k}}^2 &= \left(\|\bar{p}^*\|_{\bar{k}} - \left\|p^k\right\|_{\bar{k}}\right)\left(\|\bar{p}^*\|_{\bar{k}} + \left\|p^k\right\|_{\bar{k}}\right) \\ &\leq \left|\|\bar{p}^*\|_{\bar{k}} - \left\|p^k\right\|_{\bar{k}}\right|\left(\|\bar{p}^*\|_{\bar{k}} + \left\|p^k\right\|_{\bar{k}}\right) \\ &\leq \left\|\bar{p}^* - p^k\right\|_{\bar{k}}\left(2\|\bar{p}^*\|_{\bar{k}} + \left\|p^k - \bar{p}^*\right\|_{\bar{k}}\right) \\ &= \left\|\bar{p}^* - p^k\right\|_{\bar{k}}^2 + 2\|\bar{p}^*\|_{\bar{k}}\left\|\bar{p}^* - p^k\right\|_{\bar{k}}. \end{aligned}$$

By successively using relations (4.25) and (4.28) together again with (4.25) we get

$$\begin{aligned} \|\bar{p}^*\|_{\bar{k}}^2 - \left\|p^k\right\|_{\bar{k}}^2 &\leq \frac{2}{\gamma}\left(F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*)\right)\mathrm{e}^{-k\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} + 2\|\bar{p}^*\|_{\bar{k}}\left\|\bar{p}^* - p^k\right\|_{\bar{k}} \\ &\leq \frac{2}{\gamma}\left(F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*)\right)\mathrm{e}^{-k\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} \\ &\qquad + \frac{2\sqrt{2}}{\gamma}\left(F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*)\right)\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} \end{aligned}$$

$$\leq \left(\frac{2+2\sqrt{2}}{\gamma}\right)(F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*))\,\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} \tag{4.29}$$

Hence, continuing with relation (4.26) we get that

$$\begin{aligned} F_{\rho,\mu}(p^k) - F_{\rho,\mu}(\bar{p}^*) &\leq (F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*))\,\mathrm{e}^{-k\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} \\ &\quad + \frac{\gamma}{2}\left(\frac{2+2\sqrt{2}}{\gamma}\right)(F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*))\,\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} \\ &= (F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*))\left(\mathrm{e}^{-k\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} + (1+\sqrt{2})\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}}\right) \\ &\leq (2+\sqrt{2})\,(F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*))\,\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} \end{aligned} \tag{4.30}$$

In particular, for $k = 0$ from Corollary 4.7 we get $F_{\rho,\mu}(0) \leq F(0)$. Since p^* is the optimal solution to $(\mathrm{D}_{\mathrm{gen}})$ we further have $F(p^*) \leq F(\bar{p}^*)$. This fact together with the second inequality in (4.17) yields

$$F(p^*) - \rho D_f - \sum_{i=1}^{m} \mu_i D_{g_i} \;\leq\; F(\bar{p}^*) - \rho D_f - \sum_{i=1}^{m} \mu_i D_{g_i} \;\leq\; F_{\rho,\mu}(\bar{p}^*) \tag{4.31}$$

and

$$F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*) \;\leq\; F(0) - F(p^*) + \rho D_f + \sum_{i=1}^{m} \mu_i D_{g_i}. \tag{4.32}$$

Moreover, considering the optimal solution $\bar{p}^*$ of $(\mathrm{D}_{\rho,\mu,\gamma})$ it holds

$$\begin{aligned} F_{\rho,\mu}(\bar{p}^*) \;&\leq\; F_{\rho,\mu}(\bar{p}^*) + \frac{\gamma}{2}\,\|\bar{p}^*\|_{\bar{k}}^2 \;\leq\; F_{\rho,\mu}(p^*) + \frac{\gamma}{2}\,\|p^*\|_{\bar{k}}^2 \\ &\leq\; F(p^*) + \frac{\gamma}{2}\,\|p^*\|_{\bar{k}}^2\,. \end{aligned} \tag{4.33}$$

Using the latter inequality together with (4.17) yields

$$F_{\rho,\mu}(p^k) - F_{\rho,\mu}(\bar{p}^*) \;\geq\; F(p^k) - F(p^*) - \rho D_f - \sum_{i=1}^{m} \mu_i D_{g_i} - \frac{\gamma}{2}\,\|p^*\|_{\bar{k}}^2\,. \tag{4.34}$$

Since the lefthand side of relation (4.34) can be bounded above by taking into account relation (4.30) we have

$$F(p^k) - F(p^*) \leq F_{\rho,\mu}(p^k) - F_{\rho,\mu}(\bar{p}^*) + \rho D_f + \sum_{i=1}^{m} \mu_i D_{g_i} + \frac{\gamma}{2}\,\|p^*\|_{\bar{k}}^2$$

$$\leq (2+\sqrt{2})\left(F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*)\right) \mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}}$$
$$+ \rho D_f + \sum_{i=1}^{m} \mu_i D_{g_i} + \frac{\gamma}{2} \|p^*\|_k^2$$
$$\leq (2+\sqrt{2})\left(F_{\rho,\mu}(0) - F_{\rho,\mu}(\bar{p}^*)\right) \mathrm{e}^{-\frac{1}{2}k\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}}$$
$$+ \rho D_f + \sum_{i=1}^{m} \mu_i D_{g_i} + \frac{\gamma}{2} R^2,$$

where the last inequality is obtained by taking into account the assumed bound $R > 0$ for the norm of the optimal solution p^* of $(\mathrm{D}_{\mathrm{gen}})$. In a last step we will take into account relation (4.32) and plug this into the above inequality which finally yields

$$F(p^k) - F(p^*) \leq (2+\sqrt{2})\left(F(0) - F(p^*) + \rho D_f + \sum_{i=1}^{m} \mu_i D_{g_i}\right) \mathrm{e}^{-\frac{1}{2}k\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}}$$
$$+ \rho D_f + \sum_{i=1}^{m} \mu_i D_{g_i} + \frac{\gamma}{2} R^2, \tag{4.35}$$

and completes the proof. □

The next theorem furnishes the accuracy of the algorithmic scheme 4.1 w. r. t. the optimal objective value of $(\mathrm{D}_{\mathrm{gen}})$.

Theorem 4.9. *Let $\varepsilon > 0$ be any accuracy and F and p^* like in Proposition 4.8. Further, let $(p^k)_{k\geq 0}$ the sequences of dual iterates generated by the algorithmic scheme 4.1. Then there exists a $k' \geq 0$ such that after k' iterations it holds $F(p^{k'}) - F(p^*) \leq \varepsilon$.*

Proof. In order to achieve $\varepsilon-$accuracy we have to investigate the $m+3$ summands in the estimate (4.18) of Proposition 4.8. Our goal will be to force all summands to be lower or equal than $\frac{\varepsilon}{m+3}$. One observes that the summands ρD_f, $\frac{\gamma}{2}R^2$ and $\mu_i D_{g_i}$, $i = 1, \ldots, m$, in (4.18) do not depend on the number of iterations k. Therefore, we choose the corresponding smoothing parameters ρ, γ and μ_i, $i = 1, \ldots, m$, in view of the given accuracy $\varepsilon > 0$ to be

$$\rho(\varepsilon) = \frac{\varepsilon}{(m+3)D_f}, \quad \gamma(\varepsilon) = \frac{2\varepsilon}{(m+3)R^2} \quad \text{and} \tag{4.36}$$
$$\mu_i(\varepsilon) = \frac{\varepsilon}{(m+3)D_{g_i}}, \quad \forall i = 1, \ldots, m.$$

This yields

$$F(p^k) - F(p^*) \leq (2+\sqrt{2})\left(F(0) - F(p^*) + \frac{m+1}{m+3}\varepsilon\right) \mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} + \frac{m+2}{m+3}\varepsilon$$

where one can see that $\varepsilon-$accuracy is achieved as soon as the first term on the righthand side of the above inequality gets smaller or equal than $\frac{\varepsilon}{m+3}$ depending on the number of iterations. We obtain

$$\begin{aligned}
&(2+\sqrt{2})\left(F(0) - F(p^*) + \frac{m+1}{m+3}\varepsilon\right) \mathrm{e}^{-\frac{k'}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} \leq \frac{\varepsilon}{m+3} \\
\Leftrightarrow \quad &\frac{m+3}{\varepsilon}(2+\sqrt{2})\left(F(0) - F(p^*) + \frac{m+1}{m+3}\varepsilon\right) \leq \mathrm{e}^{\frac{k'}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} \\
\Leftrightarrow \quad &\frac{2+\sqrt{2}}{\varepsilon}\Big((m+3)(F(0) - F(p^*)) + (m+1)\varepsilon\Big) \leq \mathrm{e}^{\frac{1}{2}k'\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} \\
\Leftrightarrow \quad &2\ln\left(\frac{2+\sqrt{2}}{\varepsilon}\Big((m+3)(F(0) - F(p^*)) + (m+1)\varepsilon\Big)\right) \leq k'\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}.
\end{aligned}$$

Thus, we achieve $\varepsilon-$accuracy after

$$k' \geq 2\sqrt{\frac{L(\rho,\mu,\gamma)}{\gamma}}\ln\left(\frac{2+\sqrt{2}}{\varepsilon}\Big((m+3)(F(0) - F(p^*)) + (m+1)\varepsilon\Big)\right) \quad (4.37)$$

iterations. □

Investigating the expression under the square root we obtain by taking into consideration (4.14),

$$\frac{L(\rho,\mu,\gamma)}{\gamma} = \frac{1}{\gamma}\max_{i=1,\ldots,m}\left\{\frac{1}{\mu_i}\right\} + \frac{\sqrt{m}}{\rho\gamma}\left[\left(\sum_{i=1}^{m}\|K_i\|^2\right)\max_{i=1,\ldots,m}\{\|K_i\|^2\}\right]^{\frac{1}{2}} + 1.$$

Taking into consideration (4.36) this can be equivalently written as

$$\begin{aligned}
\frac{L(\rho,\mu,\gamma)}{\gamma} = 1 + \frac{(m+3)^2R^2}{2\varepsilon^2}\Bigg(&\max_{i=1,\ldots,m}\{D_{g_i}\} \\
&+\sqrt{m}D_f\left[\left(\sum_{i=1}^{m}\|K_i\|^2\right)\max_{i=1,\ldots,m}\{\|K_i\|^2\}\right]^{\frac{1}{2}}\Bigg)
\end{aligned} \quad (4.38)$$

and we obtain that we need $O\left(\frac{1}{\varepsilon}\ln\left(\frac{1}{\varepsilon}\right)\right)$ iterations to obtain an approximately optimal solution to $(\mathrm{D}_{\mathrm{gen}})$ within $\varepsilon-$accuracy, since $F(p^*) = -v\,(\mathrm{D}_{\mathrm{gen}})$.

4.4.2 Convergence of the gradient

In the previous subsection we analyzed the convergence properties of the fast gradient scheme w. r. t. the convergence to the optimal objective value of ($\mathrm{D}_{\mathrm{gen}}$) achieving $\varepsilon-$accuracy. Nevertheless this convergence is not sufficient to obtain an approximate primal solution, i. e. a solution to ($\mathrm{P}_{\mathrm{gen}}$). In addition, we therefore need convergence of $\left\|\nabla F_{\rho,\mu}(p^k)\right\|_{\bar{k}}$ to zero. To see this, consider the following. In Subsection 4.3.1 we observed that

$$-f_\rho^*\left(-\sum_{i=1}^m K_i^* p_i\right) = -\sup_{x\in\mathbb{R}^n}\left\{\left\langle x, -\sum_{i=1}^m K_i^* p_i\right\rangle - f(x) - \frac{\rho}{2}\|x\|^2\right\}$$

$$= \inf_{x\in\mathbb{R}^n}\left\{f(x) + \frac{\rho}{2}\left\|\frac{-\sum_{i=1}^m K_i^* p_i}{\rho} - x\right\|^2\right\} - \frac{1}{2\rho}\left\|\sum_{i=1}^m K_i^* p_i\right\|^2$$

and

$$-g_{i,\mu_i}^*(p_i) = -\sup_{x\in\mathbb{R}^{k_i}}\left\{\langle x, p_i\rangle - g_i(x) - \frac{\mu_i}{2}\|x\|_{k_i}^2\right\}$$

$$= \inf_{x\in\mathbb{R}^{k_i}}\left\{g_i(x) + \frac{\mu_i}{2}\left\|\frac{p_i}{\mu_i} - x\right\|_{k_i}^2\right\} - \frac{1}{2\mu_i}\|p_i\|_{k_i}^2$$

and that the unique minimizers of the occurring minimization problems are given by

$$x_\rho := \mathrm{Prox}_{\frac{1}{\rho}f}\left(\frac{-\sum_{i=1}^m K_i^* p_i}{\rho}\right) \quad \text{and} \quad x_{\mu_i} := \mathrm{Prox}_{\frac{1}{\mu_i}g_i}\left(\frac{p_i}{\mu_i}\right),$$

respectively, where the notations x_ρ and x_{μ_i} have been introduced to shorten notation in further calculations. Thus we get

$$f_\rho^*\left(-\sum_{i=1}^m K_i^* p_i\right) = \left\langle x_\rho, -\sum_{i=1}^m K_i^* p_i\right\rangle - f\left(x_\rho\right) - \frac{\rho}{2}\|x_\rho\|^2$$

and

$$g_{i,\mu_i}^*(p_i) = \langle x_{\mu_i}, p_i\rangle - g_i(x_{\mu_i}) - \frac{\mu_i}{2}\|x_{\mu_i}\|_{k_i}^2.$$

With this, we have for $p = (p_1, \ldots, p_m) \in \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$

$$F_{\rho,\mu}(p) = f_\rho^*\left(-\sum_{i=1}^m K_i^* p_i\right) + \sum_{i=1}^m g_{i,\mu_i}^*(p_i)$$

$$= \left\langle x_\rho, -\sum_{i=1}^{m} K_i^* p_i \right\rangle - f(x_\rho) - \frac{\rho}{2}\|x_\rho\|^2 + \sum_{i=1}^{m} \left(\langle x_{\mu_i}, p_i\rangle - g_i(x_{\mu_i}) - \frac{\mu_i}{2}\|x_{\mu_i}\|_{k_i}^2\right)$$

$$= -f(x_\rho) - \sum_{i=1}^{m} g_i(x_{\mu_i}) + \left\langle x_\rho, -\sum_{i=1}^{m} K_i^* p_i \right\rangle + \sum_{i=1}^{m} \langle x_{\mu_i}, p_i\rangle - \frac{\rho}{2}\|x_\rho\|^2 - \sum_{i=1}^{m} \frac{\mu_i}{2}\|x_{\mu_i}\|_{k_i}^2.$$

Since

$$\left\langle x_\rho, -\sum_{i=1}^{m} K_i^* p_i \right\rangle + \sum_{i=1}^{m} \langle x_{\mu_i}, p_i\rangle = \sum_{i=1}^{m} \langle x_\rho, -K_i^* p_i\rangle + \sum_{i=1}^{m} \langle x_{\mu_i}, p_i\rangle$$

$$= \sum_{i=1}^{m} \langle -K_i x_\rho, p_i\rangle + \sum_{i=1}^{m} \langle x_{\mu_i}, p_i\rangle$$

$$= \sum_{i=1}^{m} \langle x_{\mu_i} - K_i x_\rho, p_i\rangle$$

and from (4.10) we have

$$\sum_{i=1}^{m} \langle x_{\mu_i} - K_i x_\rho, p_i\rangle = \langle (x_{\mu_1} - K_1 x_\rho, \ldots, x_{\mu_m} - K_m x_\rho), (p_1, \ldots, p_m)\rangle$$

$$= \langle \nabla F_{\rho,\mu}(p), p\rangle$$

we finally get

$$f(x_\rho) + \sum_{i=1}^{m} g_i(x_{\mu_i}) = \langle \nabla F_{\rho,\mu}(p), p\rangle - \frac{\rho}{2}\|x_\rho\|^2 - \sum_{i=1}^{m} \frac{\mu_i}{2}\|x_{\mu_i}\|_{k_i}^2 - F_{\rho,\mu}(p).$$

By adding $-v(\mathrm{D_{gen}}) = F^*$ we get

$$f(x_\rho) + \sum_{i=1}^{m} g_i(x_{\mu_i}) - v(\mathrm{D_{gen}}) = \langle \nabla F_{\rho,\mu}(p), p\rangle - \frac{\rho}{2}\|x_\rho\|^2 - \sum_{i=1}^{m} \frac{\mu_i}{2}\|x_{\mu_i}\|_{k_i}^2 + (-F_{\rho,\mu}(p) - v(\mathrm{D_{gen}})) \tag{4.39}$$

and therefore

$$\left| f(x_\rho) + \sum_{i=1}^{m} g_i(x_{\mu_i}) - v(\mathrm{D_{gen}}) \right| \leq |\langle \nabla F_{\rho,\mu}(p), p\rangle| + |F_{\rho,\mu}(p) + v(\mathrm{D_{gen}})|$$

$$+ \rho D_f + \sum_{i=1}^{m} \mu_i D_{g_i}.$$

Since weak duality holds between $(\mathrm{P}_{\mathrm{gen}})$ and $(\mathrm{D}_{\mathrm{gen}})$ and we have via (4.17)

$$|F_{\rho,\mu}(p) + v\,(\mathrm{D}_{\mathrm{gen}})\,| \leq |F(p) + v\,(\mathrm{D}_{\mathrm{gen}})\,| + \rho D_f + \sum_{i=1}^{m} \mu_i D_{g_i},$$

and finally

$$f\,(x_\rho) + \sum_{i=1}^{m} g_i(x_{\mu_i}) - v\,(\mathrm{D}_{\mathrm{gen}}) \leq |\,\langle \nabla F_{\rho,\mu}(p), p\rangle\,| + |F(p) + v\,(\mathrm{D}_{\mathrm{gen}})\,| + 2\rho D_f + 2\sum_{i=1}^{m} \mu_i D_{g_i}.$$

Since we know that $F(p^k) \to -v\,(\mathrm{D}_{\mathrm{gen}})$ as k goes to infinity and the smoothing parameters ρ and μ_i, $i = 1, \ldots, m$, are chosen such that the last terms are small dependent on ε, this means that $\|\nabla F_{\rho,\mu}(p)\|_{\bar{k}}$ has to approach zero to get an approximate primal solution as from the Cauchy-Schwarz inequality we have

$$|\,\langle \nabla F_{\rho,\mu}(p), p\rangle\,| \leq \|p\|_{\bar{k}}\, \|\nabla F_{\rho,\mu}(p)\|_{\bar{k}}\,.$$

Having recognized the need of the norm of the gradient of $F_{\rho,\mu}$ going asymptotically to zero we next determine whether this is the case and, if so, we will investigate the rate of convergence. If this convergence occurred we would obtain two sequences of points

$$\left(\mathrm{Prox}_{\frac{1}{\mu_1} g_1}\left(\frac{p_1^k}{\mu_1}\right), \ldots, \mathrm{Prox}_{\frac{1}{\mu_m} g_m}\left(\frac{p_m^k}{\mu_m}\right)\right)_{k\geq 0} \subseteq \prod_{i=1}^{m} \mathrm{dom}\, g_i$$

and

$$\left(K_1\,\mathrm{Prox}_{\frac{1}{\rho} f}\left(\frac{-\sum_{i=1}^{m} K_i^* p_i^k}{\rho}\right), \ldots, K_m\,\mathrm{Prox}_{\frac{1}{\rho} f}\left(\frac{-\sum_{i=1}^{m} K_i^* p_i^k}{\rho}\right)\right)_{k\geq 0} \subseteq \prod_{i=1}^{m} K_i(\mathrm{dom}\, f)$$

whose distance gets arbitrarily small which is necessary to have a feasible solution to $(\mathrm{P}_{\mathrm{gen}})$ because all functions in its objective have to share the same argument. The following theorem provides an upper bound on the norm of the gradient of the single smoothed objective function $F_{\rho,\mu}$.

Theorem 4.10. *Let $(p^k)_{k\geq 0}$ be the sequence of dual iterates generated by the algorithmic scheme 4.1 and $F_{\rho,\mu}$ be the objective function of $(D_{\rho,\mu})$. Further, let F, R and γ be like in Proposition 4.8. Then, for any accuracy $\varepsilon > 0$ and all $k \geq 0$ it holds*

$$\left\|\nabla F_{\rho,\mu}(p^k)\right\|_{\bar{k}} \leq \left(\sqrt{L(\rho,\mu,\gamma)}+\sqrt{\gamma}\right)\sqrt{2\left(F(0)-F(p^*)+\frac{m+1}{m+3}\varepsilon\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} + \frac{\sqrt{m+2}}{m+3}\frac{2\varepsilon}{R}. \tag{4.40}$$

Proof. First, it holds

$$\left\|p^k\right\|_{\bar{k}} = \left\|p^k-\bar{p}^*+\bar{p}^*\right\|_{\bar{k}} \leq \left\|p^k-\bar{p}^*\right\|_{\bar{k}} + \left\|\bar{p}^*\right\|_{\bar{k}}, \tag{4.41}$$

where the first term on the righthand side of (4.41) can be bounded using (4.25), which yields

$$\left\|p^k\right\|_{\bar{k}} \leq \sqrt{\frac{2}{\gamma}\left(F_{\rho,\mu}(0)-F_{\rho,\mu}(\bar{p}^*)\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} + \left\|\bar{p}^*\right\|_{\bar{k}}.$$

Moreover, using (4.32) together with (4.36) we get

$$\left\|p^k\right\|_{\bar{k}} \leq \sqrt{\frac{2}{\gamma}\left(F(0)-F(p^*)+\frac{m+1}{m+3}\varepsilon\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} + \left\|\bar{p}^*\right\|_{\bar{k}}. \tag{4.42}$$

For the norm of the gradient of the single smoothed function we have

$$\begin{aligned}\left\|\nabla F_{\rho,\mu}(p^k)\right\|_{\bar{k}} &= \left\|\nabla F_{\rho,\mu}(p^k)+\gamma p^k-\gamma p^k\right\|_{\bar{k}} = \left\|\nabla F_{\rho,\mu,\gamma}(p^k)-\gamma p^k\right\|_{\bar{k}} \\ &\leq \left\|\nabla F_{\rho,\mu,\gamma}(p^k)\right\|_{\bar{k}} + \gamma\left\|p^k\right\|_{\bar{k}}.\end{aligned} \tag{4.43}$$

Next, we will bound the two summands in the above formula separately. From (2.23) we get for the norm of the gradient of the doubly regularized objective function

$$\left\|\nabla F_{\rho,\mu,\gamma}(p^k)\right\|_{\bar{k}} \leq \sqrt{2L(\rho,\mu,\gamma)\left(F_{\rho,\mu,\gamma}(p^k)-F_{\rho,\mu,\gamma}(\bar{p}^*)\right)}$$

and by applying (2.20)

$$\begin{aligned}\left\|\nabla F_{\rho,\mu,\gamma}(p^k)\right\|_{\bar{k}} &\leq \sqrt{2L(\rho,\mu,\gamma)\left(F_{\rho,\mu,\gamma}(0)-F_{\rho,\mu,\gamma}(\bar{p}^*)+\frac{\gamma}{2}\left\|\bar{p}^*\right\|_{\bar{k}}^2\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} \\ &= \sqrt{2L(\rho,\mu,\gamma)\left(F_{\rho,\mu}(0)-F_{\rho,\mu}(\bar{p}^*)\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}}.\end{aligned}$$

Finally, using (4.32) and (4.36) this becomes

$$\left\|\nabla F_{\rho,\mu,\gamma}(p^k)\right\|_{\bar{k}} \leq \sqrt{2L(\rho,\mu,\gamma)\left(F(0) - F(p^*) + \frac{m+1}{m+3}\varepsilon\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}}. \quad (4.44)$$

To get an upper bound on $\|\bar{p}^*\|_{\bar{k}}$ we notice that

$$\begin{aligned} F(p^*) + \frac{\gamma}{2}\|p^*\|_{\bar{k}}^2 &\geq F_{\rho,\mu}(p^*) + \frac{\gamma}{2}\|p^*\|_{\bar{k}} = F_{\rho,\mu,\gamma}(p^*) \\ &\geq F_{\rho,\mu,\gamma}(\bar{p}^*) = F_{\rho,\mu}(\bar{p}^*) + \frac{\gamma}{2}\|\bar{p}^*\|_{\bar{k}}^2 \\ &\geq F(\bar{p}^*) - \rho D_f - \sum_{i=1}^{m}\mu_i D_{g_i} + \frac{\gamma}{2}\|\bar{p}^*\|_{\bar{k}}^2 \\ &\geq F(p^*) - \rho D_f - \sum_{i=1}^{m}\mu_i D_{g_i} + \frac{\gamma}{2}\|\bar{p}^*\|_{\bar{k}}^2 \end{aligned}$$

which yields

$$\frac{\gamma}{2}\|\bar{p}^*\|_{\bar{k}}^2 \leq \frac{\gamma}{2}\|p^*\|_{\bar{k}}^2 + \rho D_f + \sum_{i=1}^{m}\mu_i D_{g_i},$$

and, moreover, by considering (4.36)

$$\begin{aligned} \|\bar{p}^*\|_{\bar{k}} &\leq \sqrt{\|p^*\|_{\bar{k}}^2 + \frac{2}{\gamma}\rho D_f + \frac{2}{\gamma}\sum_{i=1}^{m}\mu_i D_{g_i}} = \sqrt{\|p^*\|_{\bar{k}}^2 + \frac{2}{\gamma}\cdot\frac{m+1}{m+3}\varepsilon} \\ &= \sqrt{\|p^*\|_{\bar{k}}^2 + (m+1)R^2} \leq \sqrt{m+2}R. \end{aligned} \quad (4.45)$$

Combining (4.44), (4.42) and (4.45) by taking into consideration (4.43)

$$\begin{aligned} \left\|\nabla F_{\rho,\mu}(p^k)\right\|_{\bar{k}} &\leq \sqrt{2L(\rho,\mu,\gamma)\left(F(0) - F(p^*) + \frac{m+1}{m+3}\varepsilon\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} \\ &\quad + \gamma\sqrt{\frac{2}{\gamma}\left(F(0) - F(p^*) + \frac{m+1}{m+3}\varepsilon\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} + \gamma\sqrt{m+2}R \\ &= \left(\sqrt{L(\rho,\mu,\gamma)} + \sqrt{\gamma}\right)\sqrt{2\left(F(0) - F(p^*) + \frac{m+1}{m+3}\varepsilon\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} \\ &\quad + \gamma\sqrt{m+2}R \end{aligned}$$

$$= \left(\sqrt{L(\rho,\mu,\gamma)}+\sqrt{\gamma}\right)\sqrt{2\left(F(0)-F(p^*)+\frac{m+1}{m+3}\varepsilon\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} + \frac{\sqrt{m+2}}{m+3}\frac{2\varepsilon}{R}$$

which completes the proof. □

Next, we observe that the gradient of $F_{\rho,\mu}$ gets arbitrarily small.

Theorem 4.11. *Let $(p^k)_{k\geq 0}$ be the sequence of dual iterates generated by the algorithmic scheme 4.1 and $F_{\rho,\mu}$ be the objective function of $(D_{\rho,\mu})$. Further, let $R > 0$ such that $\|p^*\|_{\bar{k}} \leq R$. Then, for any accuracy $\varepsilon > 0$ there exists a $k' \geq 0$ such that after k' iterations*

$$\left\|\nabla F_{\rho,\mu}(p^{k'})\right\|_{\bar{k}} \leq \frac{\varepsilon}{R}. \tag{4.46}$$

Proof. We calculate the number of iterations needed to obtain this accuracy and therefore examine the inequality

$$\frac{\varepsilon}{R} \geq \left(\sqrt{L(\rho,\mu,\gamma)}+\sqrt{\gamma}\right)\sqrt{2\left(F(0)-F(p^*)+\frac{m+1}{m+3}\varepsilon\right)}\mathrm{e}^{-\frac{k'}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}} + \frac{\sqrt{m+2}}{m+3}\frac{2\varepsilon}{R}.$$

This is equivalent to

$$\frac{m+3-2\sqrt{m+2}}{m+3}\frac{\varepsilon}{R} \geq \left(\sqrt{L(\rho,\mu,\gamma)}+\sqrt{\gamma}\right)\sqrt{2\left(F(0)-F(p^*)+\frac{m+1}{m+3}\varepsilon\right)}\mathrm{e}^{-\frac{k'}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}}$$

and by further rearranging we observe

$$k' \geq 2\sqrt{\frac{L(\rho,\mu,\gamma)}{\gamma}}\ln\left(\left(\sqrt{L(\rho,\mu,\gamma)}+\sqrt{\gamma}\right)\cdot\frac{\sqrt{2(m+3)^2R^2\left(F(0)-F(p^*)+\frac{m+1}{m+3}\varepsilon\right)}}{(m+3-2\sqrt{m+2})\varepsilon}\right) \tag{4.47}$$

and get a lower bound on the number of iterations needed to achieve the desired accuracy. □

In a final step we investigate the rate of convergence w. r. t. the norm of the gradient of $F_{\rho,\mu}$. We use (4.38) and set

$$\tilde{c} := \max_{i=1,\dots,m}\{D_{g_i}\} + \sqrt{m}D_f\left[\left(\sum_{i=1}^{m}\|K_i\|^2\right)\max_{i=1,\dots,m}\left\{\|K_i\|^2\right\}\right]^{\frac{1}{2}}$$

and get with (4.36)

$$L(\rho,\mu,\gamma) = \frac{m+3}{\varepsilon}\tilde{c} + \frac{2\varepsilon}{(m+3)R^2}$$

and

$$\sqrt{\frac{L(\rho,\mu,\gamma)}{\gamma}} = \sqrt{1 + \frac{(m+3)^2R^2}{2\varepsilon^2}\tilde{c}}.$$

and therefore (4.47) can be written as

$$k' \geq 2\sqrt{\frac{(m+3)^2R^2}{2\varepsilon^2}\tilde{c}+1}\cdot\ln\left(\left(\sqrt{\frac{m+3}{\varepsilon}\tilde{c}+\frac{2\varepsilon}{(m+3)R^2}}+\sqrt{\frac{2\varepsilon}{(m+3)R^2}}\right)\cdot\frac{\sqrt{2(m+3)^2R^2\left(F(0)-F(p^*)+\frac{m+1}{m+3}\varepsilon\right)}}{(m+3-2\sqrt{m+2})\varepsilon}\right)$$

and by excluding the factor $\frac{1}{\varepsilon}$ and additional reasons

$$k' \geq \frac{2}{\varepsilon}\sqrt{\frac{(m+3)^2R^2}{2}\tilde{c}+\varepsilon^2}\left[\ln\left(\sqrt{(m+3)\tilde{c}+\frac{2\varepsilon^2}{(m+3)R^2}}+\sqrt{\frac{2\varepsilon^2}{(m+3)R^2}}\right)+\frac{3}{2}\ln\left(\frac{\sqrt[3]{2(m+3)^2R^2\left(F(0)-F(p^*)+\frac{m+1}{m+3}\varepsilon\right)}}{(m+3-2\sqrt{m+2})^{\frac{2}{3}}\varepsilon}\right)\right].$$

In conclusion we have the same rate of convergence $O\left(\frac{1}{\varepsilon}\ln\left(\frac{1}{\varepsilon}\right)\right)$ like in the considerations for the convergence of $F(p^k)$ to $F^* = -v\left(\mathrm{D}_{\mathrm{gen}}\right)$, up to a constant factor, i. e. $k = O\left(\frac{1}{\varepsilon}\ln\left(\frac{1}{\varepsilon}\right)\right)$ iterations are needed to achieve

$$F(p^k) + v\left(\mathrm{D}_{\mathrm{gen}}\right) \leq \varepsilon \qquad \text{and} \qquad \left\|\nabla F_{\rho,\mu}(p^k)\right\|_{\bar{k}} \leq \frac{\varepsilon}{R}. \tag{4.48}$$

4.5 Construction of an approximate primal solution

In this section we will show how an approximate solution to ($\mathrm{P_{gen}}$) can be obtained from the sequence $(p^k)_{k\geq 0}$ produced by the fast gradient scheme 4.1. We construct an approximate primal solution and investigate its accuracy. For this purpose we consider the sequences

$$\left(\mathrm{Prox}_{\frac{1}{\rho}f}\left(\frac{-\sum_{i=1}^m K_i^* p_i^k}{\rho}\right)\right)_{k\geq 0} \subseteq \mathrm{dom}\, f \tag{4.49}$$

and

$$\left(\mathrm{Prox}_{\frac{1}{\mu_i}g_i}\left(\frac{p_i^k}{\mu_i}\right)\right)_{k\geq 0} \subseteq \mathrm{dom}\, g_i, \qquad i=1,\ldots,m, \tag{4.50}$$

closely related to the sequence $(p^k)_{k\geq 0}$ generated by the fast gradient method. In this section we will denote by

$$x_{\rho,p^k} := \mathrm{Prox}_{\frac{1}{\rho}f}\left(\frac{-\sum_{i=1}^m K_i^* p_i^k}{\rho}\right)$$

and

$$x_{\mu_i,p^k} := \mathrm{Prox}_{\frac{1}{\mu_i}g_i}\left(\frac{p_i^k}{\mu_i}\right)$$

the proximal points of f and g_i, $i=1,\ldots,m$, respectively, at iteration $k\geq 0$. The gradient of the single smoothed objective $F_{\rho,\mu}$ can then equivalently be written as (see (4.10))

$$\nabla F_{\rho,\mu}(p^k) = \left(-K_1 x_{\rho,p^k},\ldots,-K_m x_{\rho,p^k}\right) + \left(x_{\mu_1,p^k},\ldots,x_{\mu_m,p^k}\right). \tag{4.51}$$

and we immediately obtain from (4.48) that it holds

$$\left\|\left(x_{\mu_1,p^k},\ldots,x_{\mu_m,p^k}\right) - \left(K_1 x_{\rho,p^k},\ldots,K_m x_{\rho,p^k}\right)\right\|_{\bar{k}} \leq \frac{\varepsilon}{R} \tag{4.52}$$

for k large enough. From subsection 4.4.2, equation (4.39), we have

$$f(x_{\rho,p^k}) + \sum_{i=1}^m g_i(x_{\mu_i,p^k}) - v\left(\mathrm{D_{gen}}\right) = \sum_{i=1}^m \left\langle x_{\mu_i,p^k} - K_i x_{\rho,p^k}, p_i^k\right\rangle - F_{\rho,\mu}(p^k) + F^*$$
$$- \frac{\rho}{2}\|x_{\rho,p^k}\|^2 - \sum_{i=1}^m \frac{\mu_i}{2}\|x_{\mu_i,p^k}\|_{k_i}^2$$

Depending on the accuracy $\varepsilon > 0$ that has to be determined in advance, we choose the smoothing parameters ρ and μ_i, $i = 1, \ldots, m$, according to (4.36). In the following let $k = k(\varepsilon)$ be the smallest number of iterations guaranteeing (4.37) and (4.47) which guarantees that (4.48) is true. We will now show that

$$\left| f(x_{\rho,p^k}) + \sum_{i=1}^{m} g_i(x_{\mu_i,p^k}) - v(D) \right| \leq \left(m + 3 - \sqrt{m+2} + \frac{2m+4}{m+3} \right) \varepsilon. \quad (4.53)$$

Since weak duality holds, i. e. $v\left(\mathrm{P}_{\mathrm{gen}}\right) \leq v\left(\mathrm{D}_{\mathrm{gen}}\right)$ this would imply that

$$f(x_{\rho,p^k}) + \sum_{i=1}^{m} g_i(x_{\mu_i,p^k}) \leq v\left(\mathrm{P}_{\mathrm{gen}}\right) + \left(m + 3 - \sqrt{m+2} + \frac{2m+4}{m+3} \right) \varepsilon$$

which would mean that x_{ρ,p^k} and x_{μ_i,p^k} fulfilling (4.52) and (4.53) can be interpreted as approximately optimal and feasible solutions to $(\mathrm{P}_{\mathrm{gen}})$. We will now show (4.53). On the one hand we have $F_{\rho,\mu}(p^k) - F^* \leq F(p^k) - F^* \leq \varepsilon$ and, on the other hand it holds

$$F_{\rho,\mu}(p^k) - F^* \geq F(p^k) - \rho D_f - \sum_{i=1}^{m} \mu_i D_{g_i} - F^* = F(p^k) - \frac{m+1}{m+3}\varepsilon - F^*$$

and since $F(p^k) - F^* \geq 0$ we get

$$F_{\rho,\mu}(p^k) - F^* \geq -\frac{m+1}{m+3}\varepsilon$$

and conclude that $|F_{\rho,\mu}(p^k) - F^*| \leq \varepsilon$. Now we have

$$\begin{aligned}
\left| f\left(x_{\rho,p^k}\right) + \sum_{i=1}^{m} g_i\left(x_{\mu_i,p^k}\right) - v\left(\mathrm{D}_{\mathrm{gen}}\right) \right| &\leq \left| \sum_{i=1}^{m} \left\langle x_{\mu_i,p^k} - K_i x_{\rho,p^k}, p_i^k \right\rangle \right| + \frac{\rho}{2} \left\| x_{\rho,p^k} \right\|^2 \\
&\quad + \sum_{i=1}^{m} \frac{\mu_i}{2} \|x_{\mu_i,p^k}\|_{k_i}^2 + \left| F^* - F_{\rho,\mu}\left(p^k\right) \right| \\
&\leq \left| \sum_{i=1}^{m} \left\langle x_{\mu_i,p^k} - K_i x_{\rho,p^k}, p_i^k \right\rangle \right| + \rho D_f \\
&\quad + \sum_{i=1}^{m} \mu_i D_{g_i} + \varepsilon.
\end{aligned}$$

By taking into consideration the observations in Subsection 4.4.2 we further obtain by using (4.36)

$$\rho D_f + \sum_{i=1}^{m} \mu_i D_{g_i} + \varepsilon = \frac{2m+4}{m+3}\varepsilon$$

and

$$\left|\sum_{i=1}^{m}\left\langle x_{\mu_i,p^k}-K_i x_{\rho,p^k},p_i^k\right\rangle\right|=\left|\left\langle\nabla F_{\rho,\mu}(p^k),p^k\right\rangle\right|\leq\left\|p^k\right\|_{\bar{k}}\left\|\nabla F_{\rho,\mu}(p^k)\right\|_{\bar{k}}.$$

Since we can guarantee that it holds $\left\|\nabla F_{\rho,\mu}(p^k)\right\|_{\bar{k}}\leq\frac{\varepsilon}{R}$ (cf. Theorem (4.11)) we thus get

$$\left|f\left(x_{\rho,p^k}\right)+\sum_{i=1}^{m}g_i\left(x_{\mu_i,p^k}\right)-v\left(\mathrm{D_{gen}}\right)\right|\leq\frac{\varepsilon}{R}\left\|p^k\right\|_{\bar{k}}+\frac{2m+4}{m+3}\varepsilon$$

and it remains to upper bound $\left\|p^k\right\|_{\bar{k}}$. Using (4.42) and inserting therein (4.45) we get

$$\begin{aligned}\left\|p^k\right\|_{\bar{k}}&\leq\sqrt{\frac{2}{\gamma}\left(F(0)-F(p^*)+\frac{m+1}{m+3}\varepsilon\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}}+\sqrt{m+2}R\\&=\sqrt{m+3}R\sqrt{\frac{1}{\varepsilon}\left(F(0)-F(p^*)+\frac{m+1}{m+3}\varepsilon\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}}+\sqrt{m+2}R,\end{aligned}$$

by taking into account the smoothing parameter γ (cf. (4.36)). Now, this yields

$$\left|f\left(x_{\rho,p^k}\right)+\sum_{i=1}^{m}g_i\left(x_{\mu_i,p^k}\right)-v\left(\mathrm{D_{gen}}\right)\right|\leq$$

$$\sqrt{m+3}\sqrt{\varepsilon\left(F(0)-F(p^*)+\frac{m+1}{m+3}\varepsilon\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}}+\left(\sqrt{m+2}+\frac{2m+4}{m+3}\right)\varepsilon.$$

Since k has been chosen to fulfill (4.47), i. e.

$$k\geq 2\sqrt{\frac{L(\rho,\mu,\gamma)}{\gamma}}\ln\left(\left(\sqrt{L(\rho,\mu,\gamma)}+\sqrt{\gamma}\right)\cdot\frac{\sqrt{2(m+3)^2R^2(F(0)-F(p^*)+\frac{m+1}{m+3}\varepsilon)}}{\left(m+3-2\sqrt{m+2}\right)\varepsilon}\right)$$

it also holds

$$k\geq 2\sqrt{\frac{L(\rho,\mu,\gamma)}{\gamma}}\ln\left(\frac{\sqrt{m+3}\sqrt{\varepsilon(F(0)-F(p^*)+\frac{m+1}{m+3}\varepsilon)}}{\left(m+3-2\sqrt{m+2}\right)\varepsilon}\right).\tag{4.54}$$

To see this, one has to verify that

$$\left(\sqrt{L(\rho,\mu,\gamma)}+\sqrt{\gamma}\right)\sqrt{2}(m+3)R \geq \sqrt{m+3}\sqrt{\varepsilon}$$
$$\Leftrightarrow \quad \left(\sqrt{L(\rho,\mu,\gamma)}+\sqrt{\gamma}\right)\sqrt{2}\sqrt{m+3}R \geq \sqrt{\varepsilon}.$$

From (4.36) we get

$$\varepsilon = \frac{\gamma(m+3)R^2}{2}$$

and the above relation can equivalently be written as

$$2\left(\frac{\sqrt{L(\rho,\mu,\gamma)}}{\sqrt{\gamma}}+1\right)\sqrt{\varepsilon} \geq \sqrt{\varepsilon}$$

which is immediately recognized to be fulfilled. Now (4.54) gives

$$(m+3-2\sqrt{m+2})\varepsilon \geq \sqrt{m+3}\sqrt{\varepsilon\left(F(0)-F(p^*)+\frac{m+1}{m+3}\varepsilon\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\rho,\mu,\gamma)}}}$$

and with the help of this estimate we get (4.53).

4.6 Convergence to an optimal primal solution

Recall the primal optimization problem in the representation (4.1),

$$\inf_{x\in\mathbb{R}^n}\left\{f(x)+g(Kx)\right\}.$$

In the previous section we have seen how to construct an approximate solution to this problem when solving ($\mathrm{D}_{\mathrm{gen}}$) by using the sequence produced by the fast gradient algorithm. It remains to show that in the limit, i. e. when ε approaches zero, the function f and the composition $g\circ K$ share the same argument, namely the optimal solution of ($\mathrm{P}_{\mathrm{gen}}$).

Therefore, let $(\varepsilon_n)_{n\geq 0}\subset\mathbb{R}_+$ such that $\lim_{n\to\infty}\varepsilon_n = 0$. For each $n\geq 0$ we solve the dual problem with ε_n-accuracy after $k=k(\varepsilon_n)$ iterations using the smoothing parameters $\rho(\varepsilon_n)$, $\gamma(\varepsilon_n)$ and $\mu_i(\varepsilon_n)$, $i=1,\ldots,m$. In the following we will denote by

$$\bar{x}_n = x_{\rho(\varepsilon_n),k(\varepsilon_n)} := \mathrm{Prox}_{\frac{1}{\rho(\varepsilon_n)}f}\left(-\frac{\sum_{i=1}^m K_i^* p_i^{k(\varepsilon_n)}}{\rho(\varepsilon_n)}\right)\in\mathrm{dom}\, f \quad \text{and}$$

$$\bar{y}_{n,i} = x_{\mu_i(\varepsilon_n),k(\varepsilon_n)} := \mathrm{Prox}_{\frac{1}{\mu(\varepsilon_n)}g_i}\left(\frac{p_i^{k(\varepsilon_n)}}{\mu_i(\varepsilon_n)}\right) \in \mathrm{dom}\, g_i,$$

for $i = 1, \ldots, m$ the output of the fast gradient algorithm. We set $\bar{y}_n := (\bar{y}_{n,1}, \ldots, \bar{y}_{n,m}) \in \prod_{i=1}^m \mathrm{dom}\, g_i$ and obtain points that satisfy

$$|f(\bar{x}_n) + g(\bar{y}_n) - v\,(\mathrm{D_{gen}})\,| \leq \left(m + 3 - \sqrt{m+2} + \frac{2m+4}{m+3}\right)\varepsilon_n.$$

Since $\mathrm{dom}\, f$ and $\mathrm{dom}\, g$ are bounded sets, where $\mathrm{dom}\, g = \prod_{i=1}^m \mathrm{dom}\, g_i$ the sets $\mathrm{cl}(\mathrm{dom}\, f)$ and $\mathrm{cl}(\mathrm{dom}\, g)$ are bounded sets, too, and we have for the sequences

$$(\bar{x}_n)_{n\geq 0} \subseteq \mathrm{cl}(\mathrm{dom}\, f) \quad \text{and} \quad (\bar{y}_n)_{n\geq 0} \subseteq \mathrm{cl}(\mathrm{dom}\, g).$$

Since $(\bar{x}_n)_{n\geq 0}$ and $(\bar{y}_n)_{n\geq 0}$ are bounded sequences the existence of at least one limit point for both of them can be assured and these limit points $\bar{x}$ and $\bar{y}$ lie in the sets $\mathrm{cl}(\mathrm{dom}\, f)$ and $\mathrm{cl}(\mathrm{dom}\, g)$. Furthermore, there exist subsequences that converge to these limit points. Let $(n_l)_{l\geq 0} \subseteq (n)_{n\geq 0}$ be a subset of indices such that

$$\lim_{l\to\infty} \bar{x}_{n_l} = \bar{x} \in \mathrm{cl}(\mathrm{dom}\, f) \quad \text{and} \quad \lim_{l\to\infty} \bar{y}_{n_l} = \bar{y} \in \mathrm{cl}(\mathrm{dom}\, g).$$

For every ε_{n_l} it holds

$$0 \leq \|\bar{y}_{n_l} - K\bar{x}_{n_l}\|_{\bar{k}} = \left\|\nabla F_{\rho,\mu}(p^{k(\varepsilon_{n_l})})\right\|_{\bar{k}} \leq \frac{\varepsilon_{n_l}}{R}, \tag{4.55}$$

where the right hand side of (4.55) converges to zero as $l \to \infty$ and we have

$$\lim_{l\to\infty}(\bar{y}_{n_l} - K\bar{x}_{n_l}) = \bar{y} - K\bar{x},$$

i. e. $\bar{y} = K\bar{x}$ in the limit. Taking into account the results from the previous section we now get

$$f(\bar{x}_{n_l}) + g(\bar{y}_{n_l}) \leq v\,(\mathrm{D_{gen}}) + \left(m + 3 - \sqrt{m+2} + \frac{2m+4}{m+3}\right)\varepsilon_{n_l}$$

for all $l \geq 0$. Since the functions f and g are lower semicontinuous and K is a linear operator $f(\cdot) + (g \circ K)(\cdot)$ is lower semicontinuous which yields

$$\begin{aligned} f(\bar{x}) + g(K\bar{x}) &\leq \liminf_{l\to\infty}\{f(\bar{x}_{n_l}) + g(\bar{y}_{n_l})\} \\ &\leq \liminf_{l\to\infty}\left\{v\,(\mathrm{D_{gen}}) + \left(m + 3 - \sqrt{m+2} + \frac{2m+4}{m+3}\right)\varepsilon_{n_l}\right\} \\ &= v\,(\mathrm{D_{gen}}) \leq v\,(\mathrm{P_{gen}})\,, \end{aligned}$$

i. e. we have found an element $\bar{x} \in \mathrm{cl}(\mathrm{dom}\, f)$ such that $K\bar{x} \in \mathrm{cl}(\mathrm{dom}\, g)$ and therefore $f(\bar{x}) + g(K\bar{x}) \leq v\,(\mathrm{P_{gen}})$. Since $v\,(\mathrm{P_{gen}}) < \infty$ we furthermore have $\bar{x} \in \mathrm{dom}\, f$ and $K\bar{x} \in \mathrm{dom}\, g$.

Chapter 5

Application of the double smoothing technique

In this chapter the double smoothing algorithm developed in Chapter 4 will be applied to two different tasks. First, in Section 5.1 we solve image restoration tasks for several images and different regularization functionals. It is shown that the double smoothing algorithm performs well on this problem class and outperforms other methods like FISTA. Second, in Section 5.2 we will solve the support vector regression task with the help of the double smoothing algorithm since the optimization problem that one has to solve in that case (see ($\mathrm{P_{SV}}$) in Subsection 2.2.1) fits the general setting of ($\mathrm{P_{gen}}$) after some minor reformulations.

5.1 Application to image restoration

In this section we will solve an image restoration task, i. e. we solve a regularization problem with the aim to restore an image based on an observed image that is blurred and noisy. This task is solved via the double smoothing technique and the quality of restoration measured by the improvement of the signal to noise ratio (ISNR) (see [29, 44, 1]) is compared to the well known fast iterative shrinkage-thresholding algorithm (FISTA) introduced in [7], an improved version of ISTA (cf. [32]). First, we will have a look at the different regularization problems arising in this context and how they can be solved via the double smoothing technique. After having done this, these problems are solved numerically on the basis of some famous test images.

5.1.1 Proximal points for the image restoration task

Solving the image restoration task in particular means solving an inverse problem via regularization. Denoting by $A \in \mathbb{R}^{n\times n}$ the blur operator and by $b \in \mathbb{R}^n$ an observed blurred and noisy image we want to find the unknown original image

x^* fulfilling

$$Ax = b.$$

To this end, an l_1- regularization is performed, i. e. the restored image is the minimizer of the problem

$$\inf_{x\in\mathbb{R}^n} \left\{\lambda\|x\|_1 + \|Ax - b\|^2\right\}. \tag{5.1}$$

For the motivation of considering this type of regularization for image restoration we refer to [7] and references therein.

Looking at this problem we recognize the structure of our primal problem ($\mathrm{P}_{\mathrm{gen}}$) (see Chapter 4) for $m = 1$. In order to make our double smoothing technique applicable for the image restoration task we have to impose boundedness of the domains of the functions in the objective. Since in this thesis we only consider gray scale images with pixel values between 0 for purely black pixels and 255 for purely white pixels, where all other values between 0 and 255 represent different shades of gray, this is not a restriction. Also when applying FISTA one has to solve this problem over a feasible set when the images have extreme pixel values (cf. [6]). To apply the double smoothing approach to solve the image restoration task we define the functions

$$f : \mathbb{R}^n \to \overline{\mathbb{R}}, \qquad f(x) = \lambda\|x\|_1 + \delta_S(x) \tag{5.2}$$

and

$$g_1 : \mathbb{R}^n \to \overline{\mathbb{R}}, \qquad g_1(y) = \|y - b\|^2 + \delta_S(y) \tag{5.3}$$

where n is the size of the image, i. e. the number of pixels the particular image consists of and $S = [l, u]^n \subset \mathbb{R}^n$ is the $n-$dimensional cube representing the range of the pixels. For example, the images we consider here have pixel values ranging from $l = 0$ to $u = 255$. When a scaling of these values is applied the value of u changes, naturally. Notice that the element $Ax \in S$ for all $x \in S$ since the pixels of the blurred image have the same range as the original gray scale image. The primal problem then looks

$$\inf_{x\in\mathbb{R}^n} \{f(x) + g_1(Ax)\},$$

i. e. we have recovered the structure of ($\mathrm{P}_{\mathrm{gen}}$). To solve this problem approximately we doubly smooth the corresponding dual problem ($\mathrm{D}_{\mathrm{gen}}$) and afterwards solve this doubly smoothed problem via the fast gradient algorithm. All we need to calculate in this algorithmic scheme (cf. 4.4) is the gradient of the doubly

smoothed objective function $F_{\rho,\mu,\gamma}$ at the iterate $w^k \in \mathbb{R}^n$ in each iteration $k > 0$ and the Lipschitz constant of it. Recall that the gradient is given by

$$\nabla F_{\rho,\mu,\gamma}(w^k) = \operatorname{Prox}_{\frac{1}{\mu_1} g_1}\left(\frac{w^k}{\mu_1}\right) - A \operatorname{Prox}_{\frac{1}{\rho} f}\left(-\frac{A^* w^k}{\rho}\right) + \gamma w^k. \tag{5.4}$$

(cf. 4.15) and is easily computed as soon as the occurring proximal points are provided. Thus we will now take care of them. First we calculate the proximal point of g_1 of parameter $\frac{1}{\mu_1}$ at $\frac{w^k}{\mu_1}$ and therefore consider

$$\begin{aligned}
\operatorname{Prox}_{\frac{1}{\mu_1} g_1}(z) &= \operatorname*{argmin}_{x \in \mathbb{R}^n} \left\{ g_1(x) + \frac{\mu_1}{2} \|z - x\|^2 \right\} \\
&= \operatorname*{argmin}_{x \in \mathbb{R}^n} \left\{ \|x - b\|^2 + \delta_S(x) + \frac{\mu_1}{2} \|z - x\|^2 \right\} \\
&= \operatorname*{argmin}_{x \in S} \left\{ \|x - b\|^2 + \frac{\mu_1}{2} \|z - x\|^2 \right\} \\
&= \operatorname*{argmin}_{x \in S} \left\{ \sum_{i=1}^{n} (x_i - b_i)^2 + \frac{\mu_1}{2} \sum_{i=1}^{n} (z_i - x_i)^2 \right\}.
\end{aligned}$$

To obtain the solution to the above problem we can solve n separate problems

$$\inf_{x_i \in [l,u]} \left\{ (x_i - b_i)^2 + \frac{\mu_1}{2} (z_i - x_i)^2 \right\}$$

for $i = 1, \ldots, n$. The unique minimizer of the $i-$th problem is obtained, since the objective function is differentiable, by differentiating the objective function w. r. t. x_i and setting it equal to zero. Thus, the minimizer of this problem over the whole space,

$$x_i^* = \frac{2b_i + \mu_1 z_i}{2 + \mu_1},$$

is found and, since the objective is convex, the solution of the problem restricted to the interval $[l, u]$ is obtained by projection onto this interval, i. e. applying the projection $\operatorname{Proj}_{[l,u]} : \mathbb{R} \to [l, u]$,

$$\operatorname{Proj}_{[l,u]}(x) = \begin{cases} l, & \text{if } x < l, \\ x, & \text{if } l \le x \le u, \\ u, & \text{if } x > u. \end{cases}$$

Since we need the proximal point at $\frac{w^k}{\mu_1}$ we thus get for the $i-$th component of the proximal point, by setting $z_i = \frac{(w^k)_i}{\mu_1}$,

$$\left(\text{Prox}_{\frac{1}{\mu_1}g_1}\left(\frac{w^k}{\mu_1}\right)\right)_i = \text{Proj}_{[l,u]}\left(\frac{2b_i + (w^k)_i}{2+\mu_1}\right),$$

for $i = 1, \ldots, n$, or

$$\text{Prox}_{\frac{1}{\mu_1}g_1}\left(\frac{w^k}{\mu_1}\right) = \text{Proj}_S\left(\frac{1}{2+\mu_1}(2b + w^k)\right). \tag{5.5}$$

Up to this point we thus have done half the work to calculate the gradient (5.4). It remains to calculate the corresponding proximal point of f,

$$\begin{aligned}
\text{Prox}_{\frac{1}{\rho}f}(z) &= \underset{x\in\mathbb{R}^n}{\text{argmin}}\left\{f(x) + \frac{\rho}{2}\|z-x\|^2\right\} \\
&= \underset{x\in\mathbb{R}^n}{\text{argmin}}\left\{\lambda\|x\|_1 + \delta_S(x) + \frac{\rho}{2}\|z-x\|^2\right\} \\
&= \underset{x\in S}{\text{argmin}}\left\{\lambda\|x\|_1 + \frac{\rho}{2}\|z-x\|^2\right\} \\
&= \underset{x\in S}{\text{argmin}}\left\{\sum_{i=1}^n \lambda|x_i| + \frac{\rho}{2}\sum_{i=1}^n (z_i - x_i)^2\right\}.
\end{aligned}$$

Again, we can consider n different optimization problems separately. Therefore, define the function $h : \mathbb{R} \to \mathbb{R}$,

$$h(x_i) = \lambda|x_i| + \frac{\rho}{2}(z_i - x_i)^2$$

and consider the unrestricted problem $\inf_{x_i\in\mathbb{R}}\{h(x_i)\}$. Then, x_i^* is a minimizer of this problem if and only if

$$0 \in \partial\left(\lambda|\cdot| + \frac{\rho}{2}(z_i - \cdot)^2\right)(x_i^*), \tag{5.6}$$

(e. g. see [77]). It holds that, since $\lambda > 0$ and the components are convex and the term $\frac{\rho}{2}(z_i - \cdot)$ is differentiable,

$$\partial\left(\lambda|\cdot| + \frac{\rho}{2}(z_i - \cdot)^2\right)(x_i^*) = \lambda\partial(|\cdot|)(x_i^*) - \rho z_i + \rho x_i^*$$

and therefore (5.6) can be equivalently written as

$$\rho z_i \in \lambda\partial(|\cdot|)(x_i^*) + \rho x_i^*.$$

We now have a closer look at the first term on the righthand side of the above inclusion. We have

$$\lambda\partial(|\cdot|)(x_i^*) = \begin{cases} -\lambda, & \text{if } x_i^* < 0, \\ [-\lambda, \lambda], & \text{if } x_i^* = 0, \\ \lambda, & \text{if } x_i^* > 0 \end{cases}$$

and in the following have to consider three cases. First, if $x_i^* < 0$ we have $\rho z_i = -\lambda + \rho x_i^*$, i. e.

$$x_i^* = \frac{\rho z_i + \lambda}{\rho} \quad \text{if} \quad \rho z_i < -\lambda.$$

Second, if $\rho z_i \in [-\lambda, \lambda]$ then $x_i^* = 0$. Finally, the third case yields if $x_i^* > 0$ then $\rho z_i = \lambda + \rho x_i^*$, i. e.

$$x_i^* = \frac{\rho z_i - \lambda}{\rho} \quad \text{if} \quad \rho z_i > \lambda.$$

Summarizing, we observe

$$x_i^* = \begin{cases} \frac{\rho z_i + \lambda}{\rho}, & \text{if } \rho z_i < -\lambda, \\ 0, & \text{if } -\lambda \leq \rho z_i \leq \lambda, \\ \frac{\rho z_i - \lambda}{\rho}, & \text{if } \rho z_i > \lambda, \end{cases}$$

the minimizer of $\inf_{x_i \in \mathbb{R}}\{h(x_i)\}$. For the gradient of $F_{\rho,\mu,\gamma}(w^k)$ we need the proximal point of f of parameter $\frac{1}{\gamma}$ at $-\frac{A^* w^k}{\rho}$. We thus set $z_i = -\frac{(A^* w^k)_i}{\rho}$ and obtain

$$x_i^* = \begin{cases} \frac{-(A^* w^k)_i + \lambda}{\rho}, & \text{if } (A^* w^k)_i < -\lambda, \\ 0, & \text{if } -\lambda \leq -(A^* w^k)_i \leq \lambda, \\ \frac{-(A^* w^k)_i - \lambda}{\rho}, & \text{if } -(A^* w^k)_i > \lambda \end{cases} \tag{5.7}$$

for $i = 1, \ldots, n$. The desired proximal point is then found by

$$\mathrm{Prox}_{\frac{1}{\rho} f}\left(-\frac{A^* w^k}{\rho}\right) = \mathrm{Proj}_S\left(x^*\right)$$

with $x^* = (x_1^*, \ldots, x_n^*)^{\mathrm{T}} \in \mathbb{R}^n$ given by (5.7). The above relation together with (5.5) is all it needs to calculate the gradient $\nabla F_{\rho,\mu,\gamma}(w^k)$. For the algorithmic scheme the Lipschitz constant of this gradient has to be known. This will be taken into account in the next subsection where the image deblurring and denoising task will be solved numerically.

Since we are able to solve optimization problems with a sum of more than two functions in the objective by means of the double smoothing approach, we will further consider the problem

$$\inf_{x\in\mathbb{R}^n}\{f(x)+g_1(Ax)+g_2(Ax)\} \tag{5.8}$$

in the context of the image restoration task. Here we choose the functions f and g_1 to be the same as (5.2) and (5.3), respectively, in the above considerations. Additionally, we define $g_2:\mathbb{R}^n\to\overline{\mathbb{R}}$ to be

$$g_2(y):=\|y-b\|_1+\delta_S(y). \tag{5.9}$$

The gradient of $F_{\rho,\mu,\gamma}(w^k)$ then is given by

$$\begin{aligned}\nabla F_{\rho,\mu,\gamma}(w^k)=&\left(\mathrm{Prox}_{\frac{1}{\mu_1}g_1}\left(\frac{w_1^k}{\mu_1}\right),\mathrm{Prox}_{\frac{1}{\mu_2}g_2}\left(\frac{w_2^k}{\mu_2}\right)\right)\\&-\left(A\,\mathrm{Prox}_{\frac{1}{\rho}f}\left(-\frac{\sum_{i=1}^2A^*w_i^k}{\rho}\right),A\,\mathrm{Prox}_{\frac{1}{\rho}f}\left(-\frac{\sum_{i=1}^2A^*w_i^k}{\rho}\right)\right)+\gamma w^k,\end{aligned}$$

where $w^k=(w_1^k,w_2^k)\in\mathbb{R}^n\times\mathbb{R}^n$. The component

$$\mathrm{Prox}_{\frac{1}{\mu_1}g_1}\left(\frac{w_1^k}{\mu_1}\right)=\mathrm{Proj}_S\left(\frac{1}{2+\mu_1}(2b+w_1^k)\right)$$

has already been derived in the above considerations. Furthermore, setting

$$z=-\frac{\sum_{i=1}^2A^*w_i^k}{\rho}=-\frac{1}{\rho}A^*(w_1^k+w_2^k)$$

yields

$$z_i=-\frac{1}{\rho}(A^*(w_1^k+w_2^k))_i$$

and we have (cf. (5.7))

$$x_i^*=\begin{cases}\frac{-(A^*(w_1^k+w_2^k))_i+\lambda}{\rho}, & \text{if } -(A^*(w_1^k+w_2^k))_i<-\lambda,\\ 0, & \text{if } -\lambda\le-(A^*(w_1^k+w_2^k))_i\le\lambda,\\ \frac{-(A^*(w_1^k+w_2^k))_i-\lambda}{\rho}, & \text{if } -(A^*(w_1^k+w_2^k))_i>\lambda.\end{cases} \tag{5.10}$$

This gives us the proximal point

$$\mathrm{Prox}_{\frac{1}{\rho}f}\left(-\frac{\sum_{i=1}^2A^*w_i^k}{\rho}\right)=\mathrm{Proj}_S\left(x^*\right),$$

where again $x^* = (x_1^*, \dots, x_n^*)^{\mathrm{T}} \in \mathbb{R}^n$ given by (5.10). The only work we still have to do is the calculation of the proximal point of g_2 of parameter $\frac{1}{\mu_2}$ at $z = \frac{w_2^k}{\mu_2}$, thus

$$\begin{aligned}
\mathrm{Prox}_{\frac{1}{\mu_2} g_2}(z) &= \operatorname*{argmin}_{x \in \mathbb{R}^n} \left\{ g_2(x) + \frac{\mu_2}{2} \|z - x\|^2 \right\} \\
&= \operatorname*{argmin}_{x \in \mathbb{R}^n} \left\{ \|x - b\|_1 + \delta_S(x) + \frac{\mu_2}{2} \|z - x\|^2 \right\} \\
&= \operatorname*{argmin}_{x \in S} \left\{ \|x - b\|_1 + \frac{\mu_2}{2} \|z - x\|^2 \right\} \\
&= \operatorname*{argmin}_{x \in S} \left\{ \sum_{i=1}^n |x_i - b_i| + \frac{\mu_2}{2} \sum_{i=1}^n (z_i - x_i)^2 \right\}.
\end{aligned}$$

We again separately solve n different optimization problems. Therefore, define the function $h : \mathbb{R} \to \mathbb{R}$ to be

$$h(x_i) := |x_i - b_i| + \frac{\mu_2}{2}(z_i - x_i)^2$$

and consider the unrestricted problem $\inf_{x_i \in \mathbb{R}} \{h(x_i)\}$. Then x_i^* is the unique minimizer of this problem if and only if

$$0 \in \partial \left(|\cdot - b_i| + \frac{\mu_2}{2}(z_i - \cdot)^2 \right)(x_i^*).$$

This relation can equivalently be written as

$$0 \in \partial \left(|\cdot - b_i| \right)(x_i^*) - \mu_2 z_i + \mu_2 x_i^*,$$

or, by applying [77, Theorem 2.4.2],

$$\mu_2 z_i \in \partial(|\cdot|)(x_i^* - b_i) + \mu_2 x_i^*.$$

The subdifferential in the above formula is given by

$$\partial(|\cdot|)(x_i^* - b_i) = \begin{cases} -1, & \text{if } x_i^* < b_i, \\ [-1, 1], & \text{if } x_i^* = b_i, \\ 1, & \text{if } x_i^* > b_i \end{cases}$$

and we therefore have to consider again three cases. First, if $x_i^* < b_i$ then $\mu_2 z_i = \mu_2 x_i^* - 1$. Second, if $x_i^* = b_i$ then $\mu_2 z_i \in [\mu_2 x_i^* - 1, \mu_2 x_i^* + 1]$ and third, if $x_i^* > b_i$ then $\mu_2 z_i = \mu_2 x_i^* + 1$. All in all, we have

$$x_i^* = \begin{cases} \frac{\mu_2 z_i + 1}{\mu_2}, & \text{if } \mu_2 z_i < \mu_2 b_i - 1, \\ b_i, & \text{if } \mu_2 b_i - 1 \leq \mu_2 z_i \leq \mu_2 b_i + 1, \\ \frac{\mu_2 z_i - 1}{\mu_2}, & \text{if } \mu_2 z_i > \mu_2 b_i + 1. \end{cases}$$

Since $z_i = \frac{(w_2^k)_i}{\mu_2}$ this becomes

$$x_i^* = \begin{cases} \frac{(w_2^k)_i+1}{\mu_2}, & \text{if } (w_2^k)_i < \mu_2 b_i - 1, \\ b_i, & \text{if } \mu_2 b_i - 1 \leq (w_2^k)_i \leq \mu_2 b_i + 1, \\ \frac{(w_2^k)_i-1}{\mu_2}, & \text{if } (w_2^k)_i > \mu_2 b_i + 1. \end{cases} \tag{5.11}$$

Again, by projecting $x^* = (x_1^*, \ldots, x_n^*)^{\mathrm{T}}$ given by (5.11) onto the feasible set S we obtain the desired proximal point

$$\mathrm{Prox}_{\frac{1}{\mu_2} g_2}\left(\frac{w_2^k}{\mu_2}\right) = \mathrm{Proj}_S\left(x^*\right).$$

5.1.2 Numerical results

In this section we will solve several image restoration tasks for the problems considered in the previous subsection. Our first choice will be to solve the image deblurring and denoising task of the form (5.1) for the Lena test image. After that we will use the same regularization structure for the text test image and compare the performance of our double smoothing approach to that of FISTA. This setting is the only one that we will use here that can be compared to FISTA. Finally, we apply the double smoothing technique for the cameraman test image for the setting of the regularization (5.8) in addition to

$$\inf_{x \in \mathbb{R}^n} \{\lambda \|x\|_1 + \|Ax - b\|_1\}$$

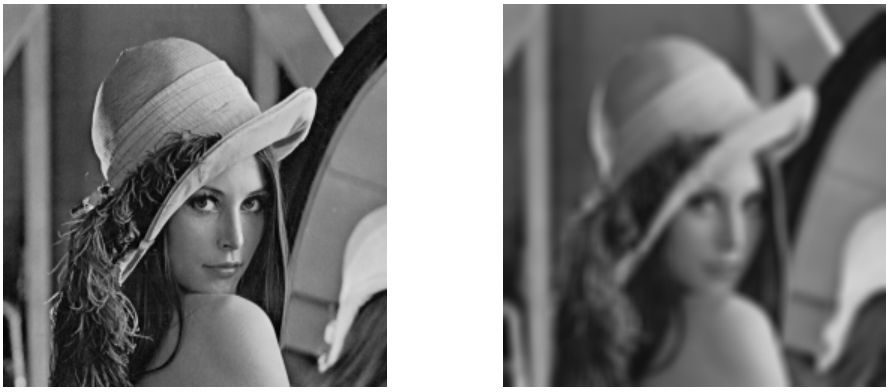
in order to compare the restoration quality for these two problems. For all of the problems the images are deblurred as follows. The operator $A : \mathbb{R}^n \to \mathbb{R}^n$ denotes the blurring operator and $b \in \mathbb{R}^n$ will be the observed blurred and noisy image. The operator A is incorporated by making use of the functions `imfilter` and `fspecial` of the MATLAB image processing toolbox. First,

```
fsp = fspecial('gaussian',9,4);
```

returns a rotationally symmetric Gaussian lowpass filter of size 9×9 with standard deviation 4. Let X be the original image. Then, for an original image X, by implementing

```
B = imfilter(X,fsp,'conv','symmetric');
```

the original image X is convolved with the filter `fsp` and returns the blurred image B. This procedure ensures that the operator $A \in \mathbb{R}^{n \times n}$, applied by using the function `imfilter` is symmetric and since each entry of the blurred image B is convex combination of elements in X and coefficients in H we have $A(S) = S$ (see [21]).



(a) original (b) blurred and noisy

Figure 5.1: The Lena test image.

The Lena test image

In a first instance we will perform the image restoration task for the famous Lena test image. This image is of size 256×256 pixels, each of them with a value within the range 0 to 255. This version of the image can for example be downloaded from *http://gtwavelet.bme.gatech.edu/images/lena.gif*. We first scale the picture such that the pixels take values in $[0, 1]$. After that we blur the image as described above and finally add a zero mean Gaussian noise with standard deviation 10^{-3}. For the double smoothing algorithm we further scale this blurred and noisy image to have pixel values in $[0, 0.1]$. In Figure 5.1 the original and the blurred image are shown.

To be able to apply the double smoothing algorithm we have to calculate the Lipschitz constant $L(\rho, \mu, \gamma)$ besides the proximal points needed for the gradient, the latter has already been done in the previous Subsection 5.1.1. The Lipschitz constant in this particular case where $m = 1$ in (4.14) is given by

$$L(\rho, \mu, \gamma) = \frac{1}{\mu_1} + \frac{1}{\rho}\|A\|^2 + \gamma. \tag{5.12}$$

For the first term we have to determine the smoothing parameter μ_1. From the third equation in (4.36) we get that

$$\mu_1 = \frac{\varepsilon}{4D_{g_1}},$$

where D_{g_1} was defined to be

$$D_{g_1} = \sup_{x \in \operatorname{dom} g_1} \left\{ \frac{1}{2}\|x\|^2 \right\}$$

in Subsection 4.4.1 (see (4.16)). In this setting the set $S = [0, 0.1]^n$ and function g_1 is given by

$$g_1(y) = \|y - b\|^2 + \delta_S(y)$$

(cf. (5.3)) we have $\operatorname{dom} g_1 = S$. Clearly, we therefore have $D_{g_1} = 327.68$ and thus, with given $\varepsilon > 0$, the smoothing parameter μ_1 can be calculated. For the second term in (5.12) we have to calculate the smoothing parameter ρ. Since $\operatorname{dom} f = S$ (see (5.2)) it turns out that $D_f = 327.68$ and with $\rho = \frac{\varepsilon}{4D_f}$ (cf. (4.36)) we obtain this smoothing parameter for a given accuracy $\varepsilon > 0$. Furthermore, we have $\|A\|^2 = 1$ (see [22]). The remaining term in (5.12), i. e. the smoothing parameter γ, is given by

$$\gamma = \frac{2\varepsilon}{4R^2},$$

(cf. (4.36)), where we set $R = 0.05$. In a last step we set $\varepsilon = 0.05$ and we have all we need to perform the image restoration task. The regularization parameter was set to $\lambda = 2\mathrm{e}^{-6}$. In Figure 5.2 one can see the resulting restored images after 10, 20, 50 and 100 iterations, respectively, together with the corresponding primal function values. The function values DS_k of primal objective in iteration k,

$$f\left(\operatorname{Prox}_{\frac{1}{\rho}f}\left(\frac{-A^*p^k}{\rho}\right)\right) + g_1\left(A\operatorname{Prox}_{\frac{1}{\rho}f}\left(\frac{-A^*p^k}{\rho}\right)\right),$$

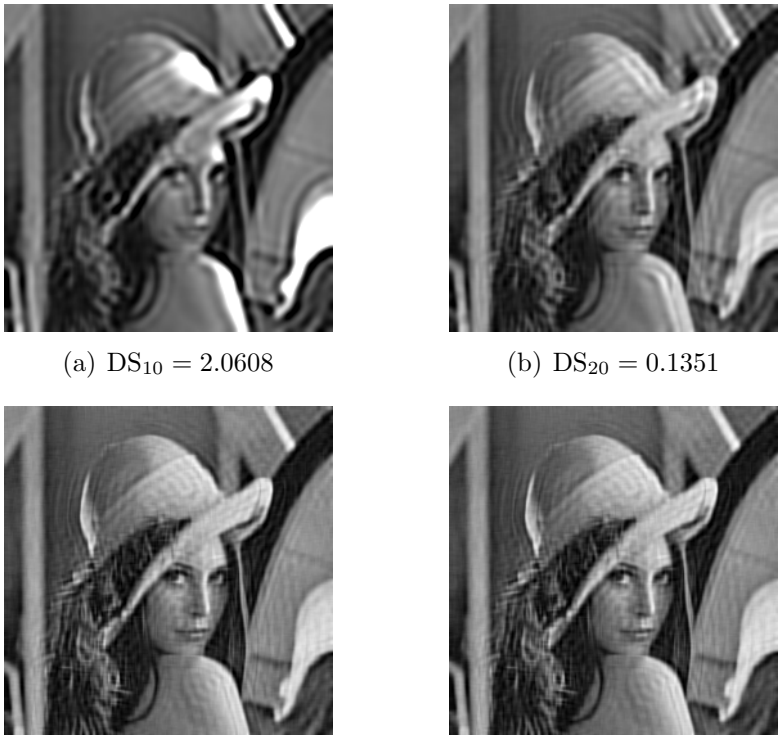
are calculated using the approximate primal solution constructed in Section 4.5. In Figure 5.3 the decrease of this primal objective value (Subfigure 5.3(a)) as well as the decrease of the norm of the gradient of the single smoothed dual objective function $F_{\rho,\mu}(p^k)$ (Subfigure 5.3(b)) is plotted dependent on the number of iterations. A Matlab implementation for this case can be found in Appendix A.1.1.

The text test image

In the following we want to compare the performance of the double smoothing algorithm with FISTA (see [7, 6]) for image restoration tasks. This is done by comparing the improvement of the signal to noise ratio (ISNR), which is defined by

$$\mathrm{ISNR}_k = 10\log_{10}\left(\frac{\|x - b\|^2}{\|x - x_k\|^2}\right),$$

where x, b and x_k denote the original image, the observed blurred and noisy image and the restored image at iteration k, respectively. This task will be done

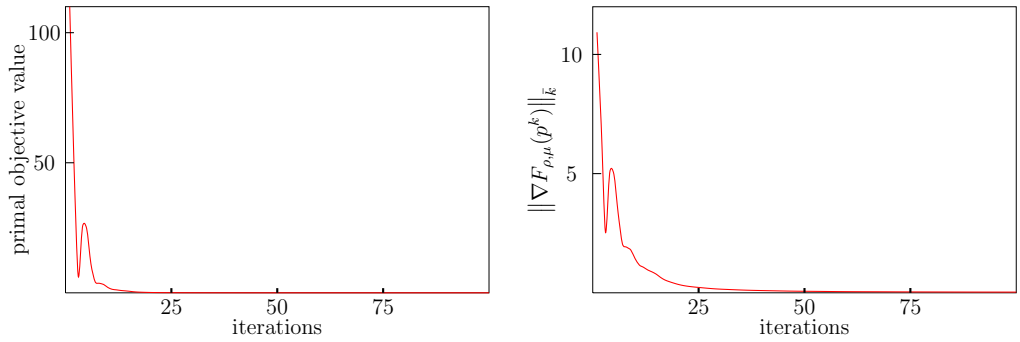


(a) $\mathrm{DS}_{10} = 2.0608$ (b) $\mathrm{DS}_{20} = 0.1351$

(c) $\mathrm{DS}_{50} = 0.0091$ (d) $\mathrm{DS}_{100} = 0.0059$

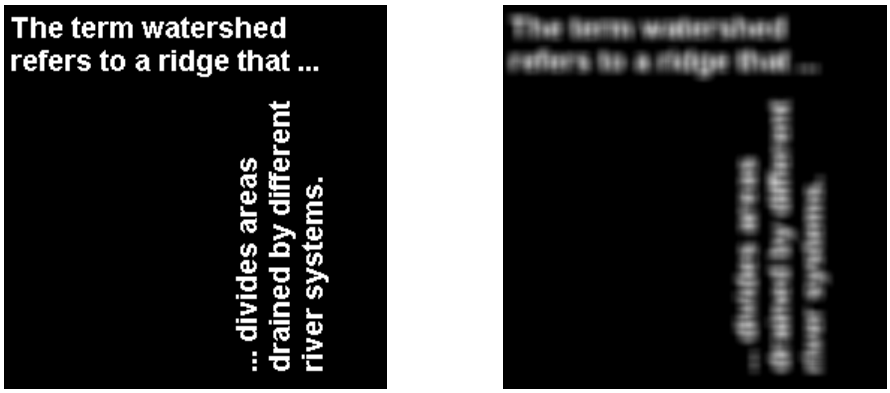
Figure 5.2: The restored Lena test image after 10, 20, 50 and 100 iterations together with the corresponding primal function values at the approximate primal solution.

by using another test image included in the MATLAB image processing toolbox, namely the image called *text*. Although we could use other objective functions we will use now the same objective function as in the considerations w. r. t. the Lena test image because these are tractable by the FISTA algorithm since the mapping $y \mapsto \|y - b\|^2$ is differentiable. The text test image is of size 256×256 pixels which take only the values 0 and 1, i. e. no gray scales are contained in the image. All pixels are just purely black or purely white. We implemented FISTA based on the feasible set $[0, 1]^n$ as proposed for such extreme situations in [7]. The image was blurred and perturbed with additional noise in the same way as the Lena test image above. The original and the blurred and noisy image can be seen in Figure 5.4.



(a) (b)

Figure 5.3: Illustrations of the decrease of the primal objective value and the norm of the gradient of $F_{\rho,\mu}(p^k)$ dependent on the number of iterations.



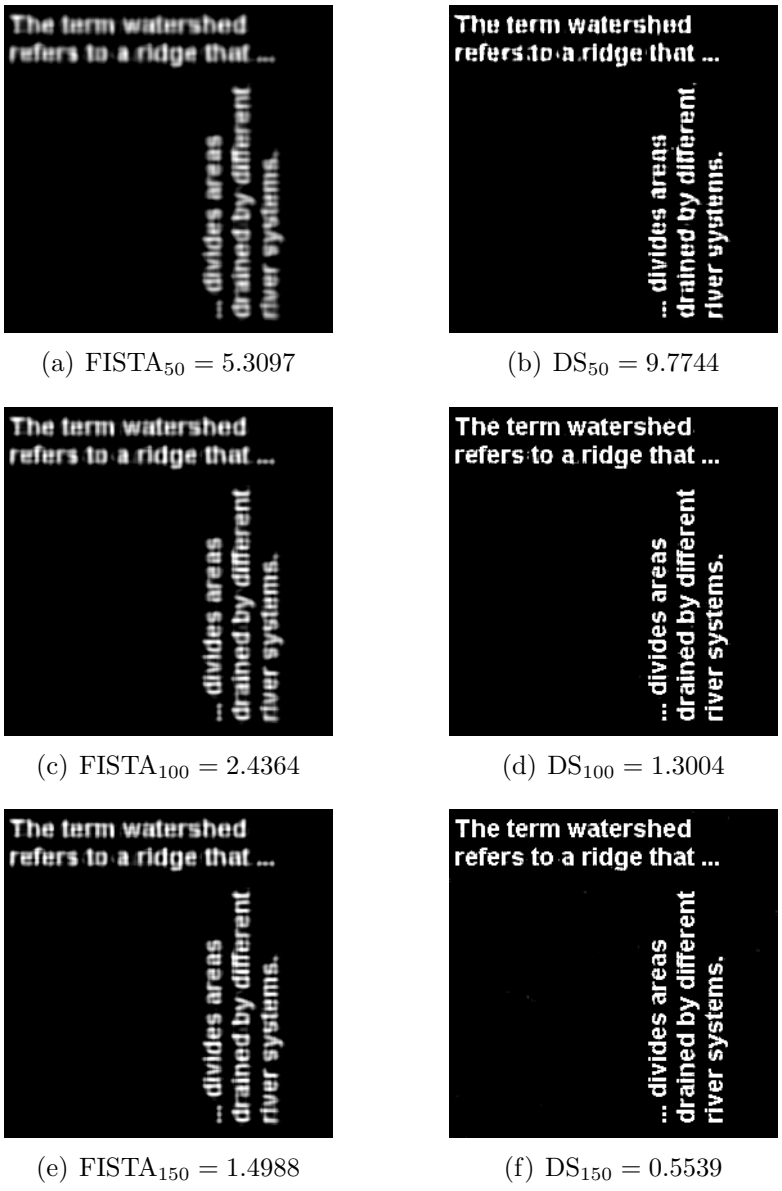
(a) original (b) blurred and noisy

Figure 5.4: The text test image.

For the double smoothing algorithm applied to the text test image we did not scale the pixels onto $[0, 0.1]$ as in the previous considerations. This image was left in its original form found in the MATLAB image processing toolbox. Therefore, it holds for the values $D_f = D_{g_1} = 32768$ where the values of the smoothing parameters μ_1 and ρ and therefore the value of Lipschitz constant of the gradient of the smooth objective functions change accordingly.

In this case we again set $\lambda = 2\text{e}^{-6}$ and $R = 0.05$. Different to the case of the Lena test image we set here $\varepsilon = 0.1$ for the accuracy of the double smoothing algorithm. In Figure 5.5 one can see the resulting images for FISTA and double smoothing after 50, 100 and 150 iterations, respectively, together with the corresponding primal function values FISTA_k and DS_k at iterations $k \in \{50, 100, 150\}$. One can see that after 150 iterations the double smoothing algorithm has produced a good approximation to the original image. A Matlab

implementation of the double smoothing approach for this problem can be found in Appendix A.1.2.



(a) $\text{FISTA}_{50} = 5.3097$ (b) $\text{DS}_{50} = 9.7744$

(c) $\text{FISTA}_{100} = 2.4364$ (d) $\text{DS}_{100} = 1.3004$

(e) $\text{FISTA}_{150} = 1.4988$ (f) $\text{DS}_{150} = 0.5539$

Figure 5.5: The restored text test image after 50, 100 and 150 iterations for both double smoothing algorithm (right column) and FISTA (left column) together with the corresponding primal function values for the approximate primal solution at the corresponding iteration.

Besides the visual verification of good performance of the double smoothing algorithm the two methods, FISTA and double smoothing, are compared by means of the SNR improvement. This is shown in Figure 5.6.

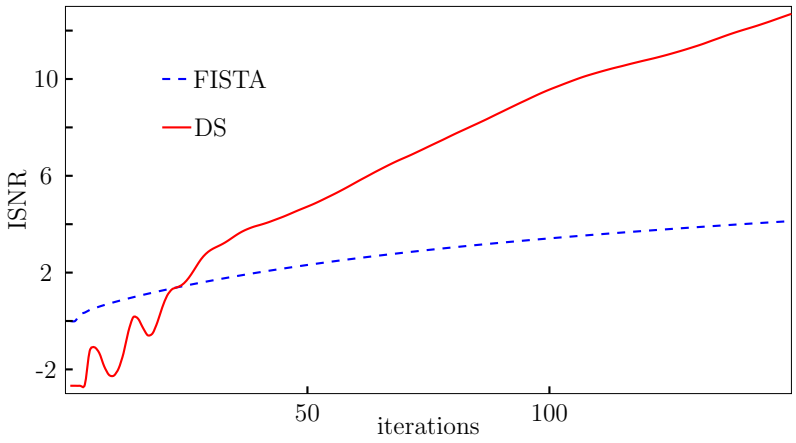


Figure 5.6: The ISNR values dependent on the number of iterations for FISTA (dashed blue line) and double smoothing (solid red line).

The cameraman test image

Finally, in a last instance we will have a look at the cameraman test image which is again a gray scale image of size 256×256 pixels. The aim now is to show that the double smoothing algorithm also works well when considering more than two functions. Another example of the use of more than two functions can be found in Section 5.2, where the double smoothing technique is applied to the support vector regression task. But for now, we restrict ourselves to two cases. The first we consider here is the l_1-regularization problem of the form

$$\inf_{x \in \mathbb{R}^n} \{\lambda \|x\|_1 + \|Ax - b\|_1\}, \tag{5.13}$$

where again A, x and b play the same role as in the considerations above in this section. Notice that this problem can not be handled via FISTA because non of the functions in the objective is differentiable. As a second example, we will show that the restoration task also works for the problem of the form (5.8). We mention again that for the double smoothing approach we add the indicator functions to each of the separate functions in the objectives of the two problems (5.13) and (5.8). The main work for applying the double smoothing technique to these problems has been done in the above calculations, namely the calculation of the different proximal points needed for the algorithmic scheme. The cameraman test image has been blurred and perturbed with additive noise like the images in the previous considerations. We now only have to care for the Lipschitz constant of the gradient of the doubly smoothed objective $F_{\rho,\mu,\gamma}$ arising from problem (5.8). The domain of all functions in the objective is the set $S = [0, 0.1]$ since we again scale the pixel value onto the range 0 to 0.1 and

therefore the Lipschitz constant is given by

$$L(\rho, \mu, \gamma) = \frac{1}{\mu_1} + \frac{2}{\rho}\|A\|^2 + \gamma,$$

(cf. (4.14)). For the computations we set $\varepsilon = 0.05$, $R = 0.05$ and $\lambda = 2\mathrm{e}^{-6}$. The solutions of the two problems in fact are very similar. That is why we only show the restored image resulting from solving

$$\inf_{x \in \mathbb{R}^n} \{f(x) + g_1(Ax) + g_2(Ax)\},$$

where

$$f(x) = \lambda\|x\|_1 + \delta_S(x), \qquad g_1(y) = \|y - b\|_1 + \delta_S(y),$$
$$\text{and} \qquad g_2(y) = \|y - b\|^2 + \delta_S(y).$$

Figure 5.7 shows the original image, the blurred and noisy image and the restored image after 200 iterations.

(a) original (b) blurred and noisy (c) restored

Figure 5.7: The cameraman test image.

In Figures 5.8, 5.9 and 5.10 the decrease of $\left\|\nabla F_{\rho,\mu}(p^k)\right\|_{\bar{k}}$, the primal objective value and the ISNR curves are shown. This example shows that the algorithm performs well when choosing more than two functions. A Matlab implementation to perform this task can be found in Appendix A.1.3.

Remark 5.1. It is worth to mention that we have seen that the double smoothing approach outperforms FISTA at least for these image restoration tasks. In fact, the considerations w. r. t. the text test image have shown that the ISNR is better for the double smoothing approach. In [22] the authors consider the same l_1-regularization task and show how the algorithm can be accelerated w. r. t. the rates of convergence of the approach (cf. Subsections 4.4.1 and 4.4.2). As

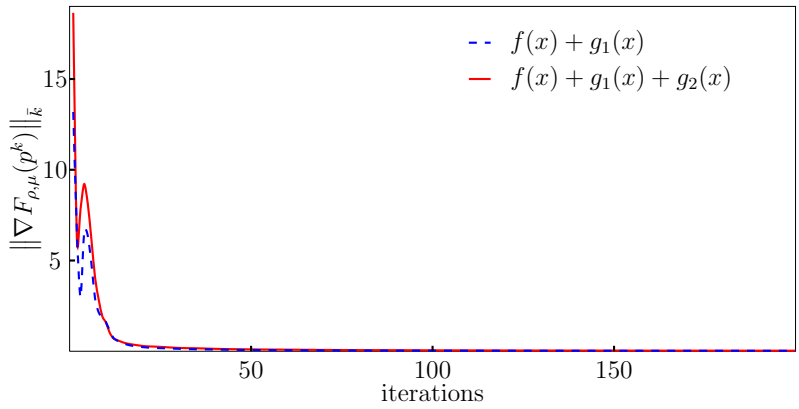


Figure 5.8: The decrease of the norm of the gradient of the single smoothed objective function $F_{\rho,\mu}$ dependent on the number of iterations for the problem with two functions (dashed blue line) and three functions (solid red line).

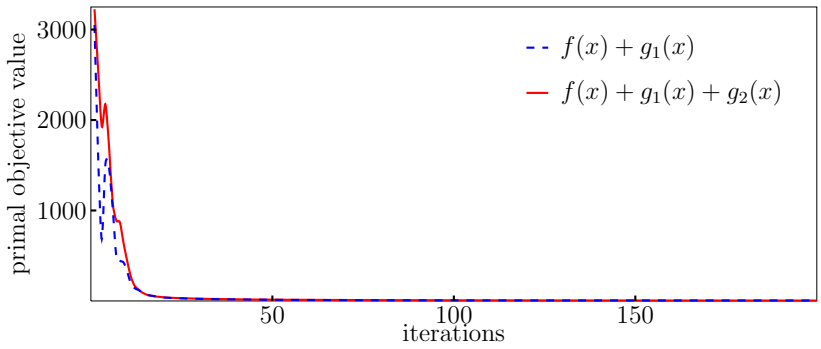


Figure 5.9: The decrease of primal objective value for the problem applying two functions (dashed blue line) and three functions (solid red line), respectively, dependent on the number of iterations.

they consider the case of the sum of two functions in the objective of the primal problem, i. e. $f(x) + g(Ax)$, several different assumptions on the functions f and g are made and the resulting improvement of the rates of convergence is shown. In the following section we will apply the double smoothing technique for solving support vector regression tasks. There, we take the function f in ($\mathrm{P}_{\mathrm{gen}}$) to be a strongly convex and continuously differentiable function. This has as consequence that the smoothing of f^* can be omitted. Nevertheless, an improvement of the rate of convergence is not obtained.

5.2 Application to support vector regression

Having seen that the double smoothing approach works well on image restoration tasks where the combination of up to three functions in the objective of the

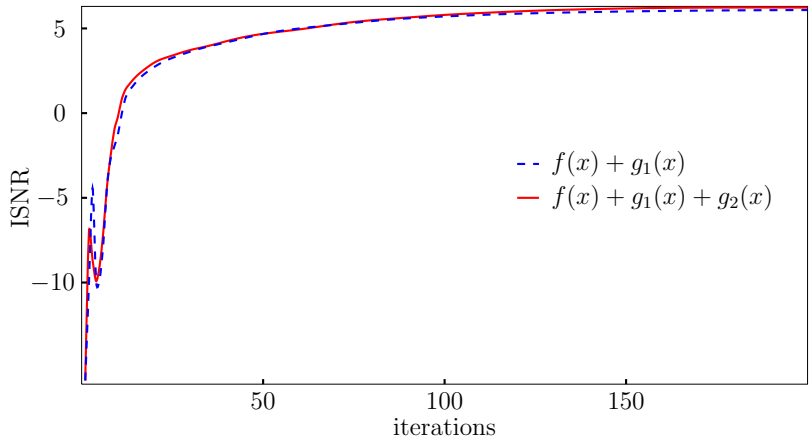


Figure 5.10: The improvement of signal to noise ratio for the case of two functions (dashed blue line) and three functions (solid red line), respectively, dependent on the number of iterations.

optimization problem ($\mathrm{P}_{\mathrm{gen}}$) is used we now want to verify the applicability of the approach for larger problems in the sense of having the sum of more than three functions in the objective. Concerning the support vector regression task we now have the sum of as much functions as there are input points available for the regression task plus the regularization term. Remember the optimization problem arising from the regression task,

$$\left(\tilde{\mathrm{P}}_{\mathrm{SV}}\right) \qquad \inf_{c\in\mathbb{R}^n} \left\{ C\sum_{i=1}^{n} v((Kc)_i, y_i) + \tilde{g}\left(\sqrt{c^{\mathrm{T}}Kc}\right) \right\}$$

derived in Chapter 3. One can see here that, where $n \in \mathbb{N}$ is the number of input data points, we will have the sum of $n+1$ different functions as objective function. In the previous subsection we were a little bit lucky in the sense that the domains of the functions in the objective could be bounded in a natural way by the range of the pixels. Considering the regression task we have to apply a slight modification to the loss function since we are forced to have the domain of the loss function bounded. In its original version this is not the case. As seen in Chapter 3 the function $\tilde{g} : \mathbb{R} \to \overline{\mathbb{R}}$ is defined as

$$\tilde{g}(t) = \begin{cases} g(t), & \text{if } t \geq 0, \\ +\infty, & \text{else,} \end{cases}$$

for a strictly monotonically increasing function $g : \mathbb{R} \to \mathbb{R}$. In this chapter we will choose the function g to be given by $g(t) = \frac{1}{2}t^2$. Therefore we have

$$\tilde{g}\left(\sqrt{c^{\mathrm{T}}Kc}\right) = g\left(\sqrt{c^{\mathrm{T}}Kc}\right) = \frac{1}{2}c^{\mathrm{T}}Kc$$

for all $c \in \mathbb{R}^n$ since K is positive semidefinite. Claiming in the following that K is even positive definite the regularization term is strongly convex with modulus $\|K\|$ and continuously differentiable and we can apply the fast gradient method by omitting the smoothing of the function f^* occurring in the dual optimization problem $(\mathrm{D}_{\mathrm{gen}})$. In particular, this means that we do not have to force the domain of f to be bounded and we will not have to modify the regularization term in a way that allows for applying the double smoothing algorithm.

5.2.1 The double smoothing technique in the case f strongly convex

In order to match the representation of the optimization problem $(\mathrm{P}_{\mathrm{gen}})$ we introduce the following by taking into account that

$$(\mathrm{P}_{\mathrm{gen}}) \qquad \inf_{c \in \mathbb{R}^n} \left\{ f(c) + \sum_{i=1}^{n} g_i(K_i c) \right\}.$$

First, let the regularization term $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ be

$$f(c) = \frac{1}{2} c^{\mathrm{T}} K c. \tag{5.14}$$

The linear operators in this setting are all identical, i. e. $K_i \equiv K$ for all $i = 1, \ldots, n$, where K is the $n \times n$ kernel matrix w. r. t. a particular set of input points $\{x_1, \ldots, x_n\} \subset \mathbb{R}^d$. In this section we will assume the kernel matrix to be positive definite. Since with these assumptions $K : \mathbb{R}^n \to \mathbb{R}^n$, we set $g_i : \mathbb{R}^n \to \overline{\mathbb{R}}$ to be

$$g_i(\cdot) = C v_i(\cdot), \tag{5.15}$$

where $v_i : \mathbb{R}^n \to \overline{\mathbb{R}}$, $v_i(z) := v(z_i, y_i)$. Obviously, $g_i(Kc) = C v((Kc)_i, y_i)$ for all $i = 1, \ldots, n$. According to Section 4.1 the dual problem reads

$$(\mathrm{D}_{\mathrm{gen}}) \qquad \sup_{(p_1, \ldots, p_n) \in \mathbb{R}^n \times \ldots \times \mathbb{R}^n} \left\{ -f^* \left(-\sum_{i=1}^{n} K^* p_i \right) - \sum_{i=1}^{n} g_i^*(p_i) \right\},$$

or, equivalently, by setting $F : \mathbb{R}^n \times \ldots \times \mathbb{R}^n \to \overline{\mathbb{R}}$,

$$F(p_1, \ldots, p_n) = f^* \left(-\sum_{i=1}^{n} K^* p_i \right) + g_i^*(p_i),$$

this can be written as

$$(\mathrm{D}_{\mathrm{gen}}) \qquad - \inf_{(p_1, \ldots, p_n) \in \mathbb{R}^n \times \ldots \times \mathbb{R}^n} \{ F(p_1, \ldots, p_n) \}.$$

In general, assuming f in $(\mathrm{P_{gen}})$ to be strongly convex, we have by [40, Theorem 4.2.1], for example, that $\operatorname{dom} f^* = \mathbb{R}^n$ and the gradient ∇f^* is Lipschitz continuous. This yields that assuming f to be strongly convex allows for omitting the requirement of $\operatorname{dom} f$ being a bounded set while still guaranteeing that the primal problem always has an optimal solution and the optimal primal and the optimal dual objective values coincide, which was the main outcome for the general case in Section 4.2. Moreover, this nice property also allows for dropping the smoothing of f^* in the first smoothing step.

First smoothing

In order to get an objective function that is continuously differentiable with Lipschitz-continuous gradient we introduce the function $F_\mu : \mathbb{R}^n \times \ldots \times \mathbb{R}^n \to \overline{\mathbb{R}}$,

$$F_\mu(p_1, \ldots, p_n) := f^*\left(-\sum_{i=1}^n K^* p_i\right) + \sum_{i=1}^n g^*_{i,\mu_i}(p_i),$$

where the functions g^*_{i,μ_i}, the approximations of g^*_i, $i = 1, \ldots, n$, are constructed in the same way as has been done in Subsection 4.3.1 (cf. (4.6)). With this we can introduce the single smoothed problem

$$(\mathrm{D}_\mu) \qquad -\inf_{(p_1,\ldots,p_n)\in\mathbb{R}^n\times\ldots\times\mathbb{R}^n} \{F_\mu(p_1, \ldots, p_n)\}$$

with an objective function being continuously differentiable by the same argumentation as in Subsection 4.3.1 and have to investigate the gradient $\nabla F_\mu(p_1, \ldots, p_n)$ and its Lipschitz constant. In this subsection we will denote by $\|\cdot\|_{\bar{n}}$ the norm in $\mathbb{R}^n \times \ldots \times \mathbb{R}^n$ defined by

$$\|x\|_{\bar{n}} = \sqrt{\|x_1\|^2 + \ldots + \|x_n\|^2}$$

for $x = (x_1, \ldots, x_n) \in \mathbb{R}^n \times \ldots \times \mathbb{R}^n$. First of all it holds

$$\nabla F_\mu(p_1, \ldots, p_n) = \nabla(f^* \circ -\bar{K}^*)(p_1, \ldots, p_n) + \left(\nabla g^*_{1,\mu_1}(p_1), \ldots, \nabla g^*_{n,\mu_n}(p_n)\right),$$

where $\bar{K}^* : \mathbb{R}^n \times \ldots \times \mathbb{R}^n \to \mathbb{R}^n$, $\bar{K}^*(p_1, \ldots, p_n) = \sum_{i=1}^n K^* p_i$ is the adjoint operator of $\bar{K} : \mathbb{R}^n \to \mathbb{R}^n \times \ldots \times \mathbb{R}^n$, $\bar{K}x = (Kx, \ldots, Kx)$. Since it holds

$$\nabla(f^* \circ -\bar{K}^*)(p_1, \ldots, p_n) = (-\bar{K}^*)^* \nabla f^*\left(-\bar{K}^*(p_1, \ldots, p_n)\right)$$

we need to calculate $\nabla f^*(y)$. Therefore, we first compute the conjugate function of $f(x) = \frac{1}{2}x^{\mathrm{T}}Kx$,

$$f^*(y) = \sup_{x\in\mathbb{R}^n} \{\langle x, y\rangle - f(x)\} = \sup_{x\in\mathbb{R}^n} \left\{\langle x, y\rangle - \frac{1}{2}x^{\mathrm{T}}Kx\right\}.$$

Since the objective in the above maximization problem is differentiable with K positive definite we get

$$f^*(y) = \frac{1}{2} y^{\mathrm{T}} K^{-1} y \tag{5.16}$$

and the gradient is easily calculated to be $\nabla f^*(y) = K^{-1}y$ and its value at point $-\bar{K}^*(p_1, \ldots, p_n)$ then is given by

$$\begin{aligned}\nabla f^*(-\bar{K}^*(p_1, \ldots, p_n)) = K^{-1}\left(-\bar{K}^*(p_1, \ldots, p_n)\right) = K^{-1}\left(-K^* \sum_{i=1}^n p_i\right) \\ = -\sum_{i=1}^n p_i\end{aligned}$$

since K is a real symmetric matrix. All in all we have found

$$\nabla(f^* \circ -\bar{K}^*)(p_1, \ldots, p_n) = -\bar{K}\left(-\sum_{i=1}^n p_i\right) = \left(K\sum_{i=1}^n p_i, \ldots, K\sum_{i=1}^n p_i\right),$$

moreover, the gradients

$$\nabla g^*_{i,\mu_i}(p_i) = \mathrm{Prox}_{\frac{1}{\mu_i} g_i}\left(\frac{p_i}{\mu_i}\right)$$

for $i = 1, \ldots, n$ have already been calculated in Subsection 4.3.1. Thus, the gradient of the single smoothed objective function in (D_μ) results in

$$\begin{aligned}\nabla F_\mu(p_1, \ldots, p_n) = \left(K\sum_{i=1}^n p_i, \ldots, K\sum_{i=1}^n p_i\right) + \\ \left(\mathrm{Prox}_{\frac{1}{\mu_1} g_1}\left(\frac{p_1}{\mu_1}\right), \ldots, \mathrm{Prox}_{\frac{1}{\mu_n} g_n}\left(\frac{p_n}{\mu_n}\right)\right).\end{aligned} \tag{5.17}$$

To determine the Lipschitz constant of this gradient we will again calculate the Lipschitz constant of each component in the above formula. First, consider for arbitrary $(p_1, \ldots, p_n), (p'_1, \ldots, p'_n) \in \mathbb{R}^n \times \ldots \times \mathbb{R}^n$ we have

$$\begin{aligned}\left\|\left(K\sum_{i=1}^n p_i, \ldots, K\sum_{i=1}^n p_i\right) - \left(K\sum_{i=1}^n p'_i, \ldots, K\sum_{i=1}^n p'_i\right)\right\|^2_{\bar{n}} \\ = \left\|\left(K\left(\sum_{i=1}^n p_i - \sum_{i=1}^n p'_i\right), \ldots, K\left(\sum_{i=1}^n p_i - \sum_{i=1}^n p'_i\right)\right)\right\|^2_{\bar{n}}\end{aligned}$$

$$= \sum_{j=1}^{n} \left\| K \sum_{i=1}^{n} (p_i - p_i') \right\|^2 = n \left\| K \sum_{i=1}^{n} (p_i - p_i') \right\|^2$$

$$\leq n\|K\|^2 \left\| \sum_{i=1}^{n} (p_i - p_i') \right\|^2 \leq n^2 \|K\|^2 \sum_{i=1}^{n} \|p_i - p_i'\|^2$$

$$= n^2 \|K\|^2 \, \|(p_1, \ldots, p_n) - (p_1', \ldots, p_n')\|_{\bar{n}}^2 \,. \tag{5.18}$$

For the second component we obtain, in analogy to the observations in Subsection 4.3.1, for arbitrary $(p_1, \ldots, p_n), (p_1', \ldots, p_n') \in \mathbb{R}^n \times \ldots \times \mathbb{R}^n$

$$\left\| \left(\mathrm{Prox}_{\frac{1}{\mu_1} g_1} \left(\frac{p_1}{\mu_1} \right), \ldots, \mathrm{Prox}_{\frac{1}{\mu_n} g_n} \left(\frac{p_n}{\mu_n} \right) \right) \right.$$
$$\left. - \left(\mathrm{Prox}_{\frac{1}{\mu_1} g_1} \left(\frac{p_1'}{\mu_1} \right), \ldots, \mathrm{Prox}_{\frac{1}{\mu_n} g_n} \left(\frac{p_n'}{\mu_n} \right) \right) \right\|_{\bar{n}}^2$$

$$\leq \max_{i=1,\ldots,n} \left\{ \frac{1}{\mu_i^2} \right\} \|(p_1, \ldots, p_n) - (p_1', \ldots, p_n')\|_{\bar{n}}^2 \,, \tag{5.19}$$

i. e. the Lipschitz constant $L(\mu)$ of $\nabla F_\mu(p_1, \ldots, p_n)$ can be summarized to

$$L(\mu) = \max_{i=1,\ldots,n} \left\{ \frac{1}{\mu_i} \right\} + n\|K\|. \tag{5.20}$$

Second smoothing

Again a second smoothing, i. e. a smoothing w. r. t. F_μ is performed to obtain an optimization problem that has a strongly convex objective function and therefore the fast gradient algorithm could be applied to it. Define $F_{\mu,\gamma} : \mathbb{R}^n \times \ldots \times \mathbb{R}^n \to \overline{\mathbb{R}}$,

$$F_{\mu,\gamma}(p_1, \ldots, p_n) = F_\mu(p_1, \ldots, p_n) + \frac{\gamma}{2} \|(p_1, \ldots, p_n)\|_{\bar{n}}^2$$

which gives rise to the doubly smoothed optimization problem

$$(\mathrm{D}_{\mu,\gamma}) \qquad - \inf_{(p_1,\ldots,p_n) \in \mathbb{R}^n \times \ldots \times \mathbb{R}^n} \{F_{\mu,\gamma}(p_1, \ldots, p_n)\}.$$

Similarly to the calculations in Subsection 4.3.2 we obtain that $\nabla F_{\mu,\gamma}$ is Lipschitz continuous with Lipschitz constant

$$L(\mu, \gamma) = L(\mu) + \gamma = \max_{i=1,\ldots,n} \left\{ \frac{1}{\mu_i} \right\} + n\|K\| + \gamma.$$

Convergence of the objective value

Now, in analogy to the calculations in Section 4.4 we could perform the convergence analysis for the convergence of $F(p^k)$ to $F^* = F(p^*)$, where $\{p^k\}_{k\geq 0} \in \mathbb{R}^n \times \ldots \times \mathbb{R}^n$, $p^k = (p_1^k, \ldots, p_n^k) \in \mathbb{R}^n \times \ldots \times \mathbb{R}^n$, is the sequence of dual variables obtained by applying the fast gradient method and p^* is the optimal solution to $(\mathrm{D}_{\mathrm{gen}})$. Further, the convergence $\left\|\nabla F_\mu(p^k)\right\|_{\bar{n}} \to 0$ will be shown. We will not perform detailed calculations here, nevertheless we state the main differences that occur in that case, i. e. when the conjugate function f^* in $(\mathrm{D}_{\mathrm{gen}})$ needs not to be approximated by a smooth function.

First of all, one has to consider F_μ and $F_{\mu,\gamma}$ instead of $F_{\rho,\mu}$ and $F_{\rho,\mu,\gamma}$, respectively. Since the smoothing of f^* is omitted, the smoothing parameter ρ vanishes and the associated value D_f needs not to be calculated (cf. (4.16)). The often used inequality (4.17) from Corollary 4.7 in the convergence analysis in this case looks

$$F_\mu(p) \leq F(p) \leq F_\mu(p) + \sum_{i=1}^n \mu_i D_{g_i} \tag{5.21}$$

for all $p = (p_1, \ldots, p_n) \in \mathbb{R}^n \times \ldots \times \mathbb{R}^n$, which is a consequence of Proposition 4.6 (ii). Notice that part (i) of this proposition does not occur in this setup. Relation (5.21) is then obtained by summing up only the inequalities stated in (ii) in Proposition 4.6 which yields

$$\sum_{i=1}^n g_{i,\mu_i}^*(p_i) \leq \sum_{i=1}^n g_i^*(p_i) \leq \sum_{i=1}^n g_{i,\mu_i}^*(p_i) + \mu_i D_{g_i}. \tag{5.22}$$

By adding the term $f^*\left(-\sum_{i=1}^n K^* p_i\right)$ this results in (5.21). Going further, relation (4.18) then gets

$$F(p^k) - F(p^*) \leq (2+\sqrt{2})\left(F(0) - F(p^*) + \sum_{i=1}^n \mu_i D_{g_i}\right) \mathrm{e}^{-\frac{1}{2}k\sqrt{\frac{\gamma}{L(\mu,\gamma)}}} \tag{5.23}$$

$$+ \sum_{i=1}^n \mu_i D_{g_i} + \frac{\gamma}{2} R^2. \tag{5.24}$$

In order to achieve $\varepsilon-$accuracy, i. e. $F(p^k) - F(p^*) \leq \varepsilon$ one now has to force the sum of the $n+2$ terms in (5.23) to be less than or equal ε, i. e. each summand has to be less than or equal $\frac{\varepsilon}{n+2}$. For the latter $n+1$ terms in (5.23) this is achieved by setting the smoothing parameters to

$$\mu_i(\varepsilon) = \frac{\varepsilon}{(n+2)D_{g_i}}, \quad \forall i = 1, \ldots, n, \quad \text{and} \quad \gamma(\varepsilon) = \frac{2\varepsilon}{(n+2)R^2}. \tag{5.25}$$

In view of that choice we have

$$F(p^k) - F(p^*) \leq (2+\sqrt{2})\left(F(0) - F(p^*) + \frac{n\varepsilon}{n+2}\right)\mathrm{e}^{-\frac{1}{2}k\sqrt{\frac{\gamma}{L(\mu,\gamma)}}} + \frac{n+1}{n+2}\varepsilon$$

and achieving $\varepsilon-$accuracy depends on the number of iterations needed until it holds

$$(2+\sqrt{2})\left(F(0) - F(p^*) + \frac{n\varepsilon}{n+2}\right)\mathrm{e}^{-\frac{1}{2}k\sqrt{\frac{\gamma}{L(\mu,\gamma)}}} \leq \frac{\varepsilon}{n+2}$$

$$\Leftrightarrow \quad \frac{n+2}{\varepsilon}(2+\sqrt{2})\left(F(0) - F(p^*) + \frac{n}{n+2}\varepsilon\right) \leq \mathrm{e}^{\frac{k}{2}\sqrt{\frac{\gamma}{L(\mu,\gamma)}}}$$

$$\Leftrightarrow \quad 2\ln\left(\frac{(2+\sqrt{2})}{\varepsilon}\Big((n+2)(F(0) - F(p^*)) + n\varepsilon\Big)\right) \leq k\sqrt{\frac{\gamma}{L(\mu,\gamma)}},$$

i. e. after

$$k \geq 2\sqrt{\frac{L(\mu,\gamma)}{\gamma}}\ln\left((2+\sqrt{2})\left(\frac{n+2}{\varepsilon}(F(0) - F(p^*)) + n\right)\right)$$

iterations the desired accuracy is achieved. We now further investigate the term with the square root in the above formula. We have

$$\begin{aligned}\frac{L(\mu,\gamma)}{\gamma} &= \frac{1}{\gamma}\max_{i=1,\ldots,n}\left\{\frac{1}{\mu_i}\right\} + \frac{n}{\gamma}\|K\| + 1 \\ &= \frac{(n+2)^2R^2}{2\varepsilon^2}\max_{i=1,\ldots,n}\{D_{g_i}\} + \frac{n(n+2)^2R^2}{2\varepsilon}\|K\| + 1,\end{aligned}$$

and by taking into account (5.25), which in conclusion means that we obtain the same rate of convergence as in Subsection 4.4.1, namely we need $O(\frac{1}{\varepsilon}\ln\left(\frac{1}{\varepsilon}\right))$ iterations to achieve $\varepsilon-$accuracy for the optimal objective value of $(\mathrm{D}_{\mathrm{gen}})$.

Convergence of the gradient

In a next step we will have a closer look at the convergence of the gradient of $F_\mu(p^k)$ to zero as k approaches infinity in the case where f is strongly convex and continuously differentiable. First, like in Subsection 4.4.2 notice that by denoting

$$x_{\mu_i,p^k} := \mathrm{Prox}_{\frac{1}{\mu_i}g_i}\left(\frac{p_i^k}{\mu_i}\right)$$

the proximal point of g_i of parameter $\frac{1}{\mu_i}$ at $\frac{p_i^k}{\mu_i}$,

$$g_{i,\mu_i}^*(p_i^k) = \left\langle x_{\mu_i,p^k}, p_i \right\rangle - g_i\left(x_{\mu_i,p^k}\right) - \frac{\mu_i}{2}\left\|x_{\mu_i,p^k}\right\|^2$$

for all $i = 1, \ldots, n$. With that observation and the fact that we do not apply an approximation of f^* here, we get

$$\begin{aligned} F_\mu(p^k) &= f^*\left(-\sum_{i=1}^n K^* p_i^k\right) + \sum_{i=1}^n g_{i,\mu_i}^*(p_i^k) \\ &= f^*\left(-\sum_{i=1}^n K^* p_i^k\right) + \sum_{i=1}^n \left(\left\langle x_{\mu_i,p^k}, p_i^k \right\rangle - g_i\left(x_{\mu_i,p^k}\right) - \frac{\mu_i}{2}\left\|x_{\mu_i,p^k}\right\|^2\right). \end{aligned} \tag{5.26}$$

Since we have calculated the conjugate function f^* of f we can now calculate the conjugate of f at point $-\sum_{i=1}^n K^* p_i^k$,

$$\begin{aligned} f^*\left(-\sum_{i=1}^n K^* p_i^k\right) &= \frac{1}{2}\left(-\sum_{i=1}^n K^* p_i^k\right)^{\mathrm{T}} K^{-1}\left(-\sum_{i=1}^n K^* p_i^k\right) \\ &= \frac{1}{2}\left(-\sum_{i=1}^n p_i^k\right)^{\mathrm{T}} (K^*)^{\mathrm{T}} K^{-1} K^*\left(-\sum_{i=1}^n p_i^k\right) \\ &= \frac{1}{2}\left(-\sum_{i=1}^n p_i^k\right)^{\mathrm{T}} K\left(-\sum_{i=1}^n p_i^k\right), \end{aligned}$$

since the kernel matrix K is a real symmetric matrix and therefore $K^* = K^{\mathrm{T}}$. With this observation (5.26) can be continued,

$$\begin{aligned} F_\mu(p^k) &= \frac{1}{2}\left\langle -\sum_{i=1}^n p_i^k, K\left(-\sum_{i=1}^n p_i^k\right)\right\rangle + \sum_{i=1}^n \left(\left\langle x_{\mu_i,p^k}, p_i\right\rangle - g_i\left(x_{\mu_i,p^k}\right)\right. \\ &\quad \left. -\frac{\mu_i}{2}\left\|x_{\mu_i,p^k}\right\|^2\right) \\ &= -\frac{1}{2}\left\langle -\sum_{i=1}^n p_i^k, K\left(-\sum_{i=1}^n p_i^k\right)\right\rangle + \left\langle \sum_{i=1}^n p_i^k, K\sum_{i=1}^n p_i^k\right\rangle \\ &\quad + \sum_{i=1}^n \left\langle x_{\mu_i,p^k}, p_i^k\right\rangle - \sum_{i=1}^n g_i\left(x_{\mu_i,p^k}\right) - \sum_{i=1}^n \frac{\mu_i}{2}\left\|x_{\mu_i,p^k}\right\|^2 \end{aligned}$$

$$= -f\left(-\sum_{i=1}^{n} p_i^k\right) - \sum_{i=1}^{n} g_i\left(x_{\mu_i,p^k}\right) + \left\langle \sum_{i=1}^{n} p_i^k, K\sum_{i=1}^{n} p_i^k \right\rangle$$

$$+ \sum_{i=1}^{n} \left\langle x_{\mu_i,p^k}, p_i^k \right\rangle - \sum_{i=1}^{n} \frac{\mu_i}{2} \left\| x_{\mu_i,p^k} \right\|^2.$$

Taking into consideration that

$$\left\langle \sum_{i=1}^{n} p_i^k, K\sum_{i=1}^{n} p_i^k \right\rangle + \sum_{i=1}^{n} \left\langle x_{\mu_i,p^k}, p_i^k \right\rangle = \sum_{i=1}^{n} \left\langle p_i^k, x_{\mu_i,p^k} + K\sum_{j=1}^{n} p_j^k \right\rangle$$

$$= \left\langle p^k, \nabla F_\mu(p^k) \right\rangle$$

we further get

$$F_\mu(p^k) = -f\left(-\sum_{i=1}^{n} p_i^k\right) - \sum_{i=1}^{n} g_i\left(x_{\mu_i,p^k}\right) + \left\langle p^k, \nabla F_\mu(p^k) \right\rangle - \sum_{i=1}^{n} \frac{\mu_i}{2} \left\| x_{\mu_i,p^k} \right\|^2.$$

By rearranging the last formula and by adding $-v\left(\mathrm{D}_{\mathrm{gen}}\right) = F^*$ we observe that, similarly to the observations in Subsection 4.4.2,

$$f\left(-\sum_{i=1}^{n} p_i^k\right) + \sum_{i=1}^{n} g_i\left(x_{\mu_i,p^k}\right) - v\left(\mathrm{D}_{\mathrm{gen}}\right) = \left\langle p^k, \nabla F_\mu(p^k) \right\rangle - \sum_{i=1}^{n} \frac{\mu_i}{2} \left\| x_{\mu_i,p^k} \right\|^2$$
$$- F_\mu(p^k) + F^*. \quad (5.27)$$

Taking the absolute value on both sides yields

$$\left| f\left(-\sum_{i=1}^{n} p_i^k\right) + \sum_{i=1}^{n} g_i\left(x_{\mu_i,p^k}\right) - v\left(\mathrm{D}_{\mathrm{gen}}\right) \right| \leq \left| \left\langle p^k, \nabla F_\mu(p^k) \right\rangle \right| + \sum_{i=1}^{n} \frac{\mu_i}{2} \|x_{\mu_i,p^k}\|^2$$
$$+ \left| F_\mu(p^k) - F^* \right|,$$

indicating by using the Cauchy-Schwarz inequality as done in Subsection 4.4.2 $\left| \left\langle p^k, \nabla F_\mu(p^k) \right\rangle \right| \leq \left\| p^k \right\|_{\bar{n}} \left\| \nabla F_\mu(p^k) \right\|_{\bar{n}}$ that the norm of the gradient of $F_\mu(p^k)$ has to approach zero. Therefore, for the sequence of dual variables $(p^k)_{k\geq 0}$, where again $p^k = (p_1^k, \ldots, p_n^k) \in \mathbb{R}^n \times \ldots \times \mathbb{R}^n$, obtained by applying the fast gradient method, we have already seen, that for k big enough we achieve $\varepsilon-$accuracy for the objective value of the single smoothed problem (D_μ). In addition, account for (5.25) and obtain

$$\left| f\left(-\sum_{i=1}^{n} p_i^k\right) + \sum_{i=1}^{n} g_i\left(x_{\mu_i,p^k}\right) - v\left(\mathrm{D}_{\mathrm{gen}}\right) \right| \leq \left\| p^k \right\|_{\bar{n}} \left\| \nabla F_\mu(p^k) \right\|_{\bar{n}} + \frac{2n+2}{n+2}\varepsilon.$$

Now we will show that the gradient approaches zero on the limit but will not go into detail because the calculations are analogous as in Subsection 4.4.2. Nevertheless, the occurring differences will be mentioned here. The starting point for the convergence analysis is, similar to (4.43),

$$\begin{aligned}\left\|\nabla F_\mu(p^k)\right\|_{\bar{n}} &= \left\|\nabla F_\mu(p^k) + \gamma p^k - \gamma p^k\right\|_{\bar{n}} = \left\|\nabla F_{\mu,\gamma}(p^k) - \gamma p^k\right\|_{\bar{n}} \\ &\leq \left\|\nabla F_{\mu,\gamma}(p^k)\right\|_{\bar{n}} + \gamma \left\|p^k\right\|_{\bar{n}}. \end{aligned} \tag{5.28}$$

It turns out that

$$\left\|\nabla F_{\mu,\gamma}(p^k)\right\|_{\bar{n}} \leq \sqrt{2L(\mu,\gamma)\left(F(0) - F(p^*) + \frac{n}{n+2}\varepsilon\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\mu,\gamma)}}}$$

and

$$\|\bar{p}^*\|_{\bar{n}} \leq \sqrt{\|p^*\|_{\bar{n}}^2 + nR^2} \leq \sqrt{(n+1)}R,$$

i. e. the two summands in (5.28) are bounded above which thus yields

$$\begin{aligned}\left\|\nabla F_\mu(p^k)\right\|_{\bar{n}} \leq &\sqrt{2L(\mu,\gamma)\left(F(0) - F(p^*) + \frac{n}{n+2}\varepsilon\right)}\mathrm{e}^{-\frac{k}{2}\sqrt{\frac{\gamma}{L(\mu,\gamma)}}} \\ &+ \frac{\sqrt{n+1}}{n+2}\frac{2\varepsilon}{R},\end{aligned}$$

where the expression on the right hand side can not get lower than the last summand. We therefore will require an accuracy of

$$\left\|\nabla F_\mu(p^k)\right\|_{\bar{n}} \leq \frac{\varepsilon}{R}$$

for the whole expression. One could go on and perform the calculations that give a lower bound on the number of iterations needed to achieve $\varepsilon-$accuracy up to a constant factor and would obtain the same rate of convergence as before, namely $k = O\left(\frac{1}{\varepsilon}\ln\left(\frac{1}{\varepsilon}\right)\right)$ iterations are needed to obtain

$$F(p^k) - F(p^*) \leq \varepsilon \qquad \text{and} \qquad \left\|\nabla F_\mu(p^k)\right\|_{\bar{n}} \leq \frac{\varepsilon}{R}.$$

Using the above estimates one can show, similar like in Section 4.5, that it holds

$$f\left(-\sum_{i=1}^{n} p_i^k\right) + \sum_{i=1}^{n} g_i\left(x_{\mu_i,p^k}\right) - v\left(\mathrm{D}_{\mathrm{gen}}\right) \leq \left(n + 2 - \sqrt{n+1} + \frac{2n+2}{n+2}\right)\varepsilon. \tag{5.29}$$

Convergence to an approximate optimal solution

It remains to show that an approximate optimal and feasible primal solution exists, i. e. that in the limit $\varepsilon \to 0$ the functions f and $g_i \circ K$ share the same argument.

Therefore, we proceed similar like in Section 4.6, while some steps have to be modified. Let $(\varepsilon_t)_{t\geq 0} \subset \mathbb{R}$, $\varepsilon_t > 0$ for all $t \geq 0$, be a decreasing sequence such that $\lim_{t\to\infty} \varepsilon_t = 0$. For each $t \geq 0$ we have seen that we have to perform $k = k(\varepsilon_t)$ iterations of the fast gradient algorithm with smoothing parameters $\mu_i(\varepsilon_t)$, $i = 1, \ldots, n$, and $\gamma(\varepsilon_t)$ (cf. (5.25)) such that it holds

$$F(p^{k(\varepsilon_t)}) - F(p^*) \leq \varepsilon_t \qquad \text{and} \tag{5.30}$$

$$\left\|\nabla F_{\mu(\varepsilon_t)}(p^{k(\varepsilon_t)})\right\|_{\bar{n}} \leq \frac{\varepsilon_t}{R}, \tag{5.31}$$

where $p^{k(\varepsilon_t)} = (p_1^{k(\varepsilon_t)}, \ldots, p_n^{k(\varepsilon_t)}) \in \mathbb{R}^n \times \ldots \times \mathbb{R}^n$. Now we denote by

$$\bar{x}_t := -\sum_{i=1}^{n} p_i^{k(\varepsilon_t)} \in \operatorname{dom} f = \mathbb{R}^n \qquad \text{and}$$

$$\bar{y}_{i,t} := x_{\mu_1(\varepsilon_t),k(\varepsilon_t)} \in \operatorname{dom} g_i, \qquad \forall i = 1, \ldots, n,$$

where

$$x_{\mu_i(\varepsilon_t),k(\varepsilon_t)} := \operatorname{Prox}_{\frac{1}{\mu_i} g_i}\left(\frac{p_i^{k(\varepsilon_t)}}{\mu_i}\right),$$

$i = 1, \ldots, n$. From (5.17) we thus have by accounting for (5.31)

$$0 \leq \|(\bar{y}_{1,t}, \ldots, \bar{y}_{n,t}) - (K\bar{x}_t, \ldots, K\bar{x}_t)\|_{\bar{n}} \leq \frac{\varepsilon_t}{R}. \tag{5.32}$$

Now, since $\operatorname{dom} g_i$ is a bounded set the sequence $(\bar{y}_{i,t})_{t\geq 0}$ is a bounded sequence for all $i = 1, \ldots, n$ and there exists a subsequence $(\bar{y}_{i,t_l})_{l\geq 0} \subseteq (\bar{y}_{i,t})_{t\geq 0}$ such that $\lim_{l\to\infty} \bar{y}_{i,t_l} = \bar{y}_i \in \operatorname{cl}(\operatorname{dom} g_i)$. Then, by taking into account (5.32) the sequence $(K\bar{x}_t)_{t\geq 0}$ is a bounded sequence which has a subsequence $(K\bar{x}_{t_l})_{l\geq 0}$ that converges to a point in $\mathbb{R}^n$ as $l \to \infty$. Since K is invertible, we get that $(\bar{x}_{t_l})_{l\geq 0} \to \bar{x}$ and we have $K\bar{x} = \bar{y}_i$ for all $i = 1, \ldots, n$. Now, taking into account (5.29) we have

$$f(\bar{x}_{t_l}) + \sum_{i=1}^{n} g_i\left(\bar{y}_{i,t_l}\right) \leq v\left(\mathrm{D}_{\mathrm{gen}}\right) + \left(n + 2 - \sqrt{n+1} + \frac{2n+2}{n+2}\right)\varepsilon_{t_l}$$

for all $l \geq 0$. Again, like at the end of Section 4.6 we get by taking into account the lower semicontinuity of f and g_i, $i = 1, \ldots, n$,

$$\begin{aligned} f(\bar{x}) + \sum_{i=1}^{n} g_i(K\bar{x}) &\leq \liminf_{l\to\infty} \left\{ f(\bar{x}_{t_l}) + \sum_{i=1}^{n} g_i\left(\bar{y}_{i,t_l}\right) \right\} \\ &\leq \liminf_{l\to\infty} \left\{ v\left(\mathrm{D}_{\mathrm{gen}}\right) + \left(n + 2 - \sqrt{n+1} + \frac{2n+2}{n+2} \right) \varepsilon_{t_l} \right\} \\ &= v\left(\mathrm{D}_{\mathrm{gen}}\right) \leq v\left(\mathrm{P}_{\mathrm{gen}}\right) < \infty. \end{aligned}$$

We have found an element $\bar{x} \in \operatorname{dom} f$ and $K\bar{x} \in \operatorname{dom} g_i$, $i = 1, \ldots, n$, and $\bar{x}$ is an optimal solution to $(\mathrm{P}_{\mathrm{gen}})$.

5.2.2 Proximal points for different loss functions

In this subsection we will calculate the proximal points that are needed for the computation of the gradient that occurs in the algorithmic scheme employed by the double smoothing technique for different choices of the loss function. In particular, we can solve this regression task via the fast gradient algorithm due to the properties of the function f being strongly convex and continuously differentiable with Lipschitz continuous gradient. If this would not be the case, we would have to require the domain of f to be bounded which would mean a restriction on the variable space. Nevertheless, we still have to require the domain of the functions $g_i(\cdot) = Cv_i(\cdot)$ (cf. (5.15)) to be bounded since these functions are not differentiable in general. Especially in the regression task this can be motivated.

The $\varepsilon-$insensitive loss function

In the following we will consider the $\varepsilon-$insensitive loss function $v_\varepsilon : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ defined by

$$v_\varepsilon(a, y) = |a - y|_\varepsilon := \max\{0, |a - y| - \varepsilon\} = \begin{cases} 0, & \text{if } |a - y| \leq \varepsilon, \\ |a - y| - \varepsilon, & \text{else.} \end{cases}$$

Using this as the loss function the functions g_i in the formulation of $(\mathrm{P}_{\mathrm{gen}})$ are given by

$$g_i(K_i x) = g_i(Kx) = Cv_\varepsilon\big((Kx)_i, y_i\big) = Cv_i(Kx),$$

where $v_i(z) = v_\varepsilon(z_i, y_i)$ for all $i = 1, \ldots, n$. As mentioned before, these functions need to have bounded domain. Therefore, for $\xi \geq \varepsilon > 0$ define the set

$$\Lambda := [y_1 - \xi, y_1 + \xi] \times \ldots \times [y_n - \xi, y_n + \xi] \subset \mathbb{R} \times \ldots \times \mathbb{R} \tag{5.33}$$

and consider the functions $g_i : \mathbb{R}^n \to \overline{\mathbb{R}}$, $i = 1, \ldots, n$, to be used in the primal problem ($\mathrm{P}_{\mathrm{gen}}$),

$$g_i(\cdot) = Cv_i(\cdot) = C|\langle e_i, \cdot\rangle - y_i|_\varepsilon + \delta_\Lambda(\cdot)$$

which have bounded domain. These changes w. r. t. the loss function will force the function value $(Kx)_i$, $i = 1, \ldots, n$, of the regression function at point $x \in X$ to lie within the $\xi-$tube around the given sample y_i. For ξ large enough the behavior of the regression function will be identical to that observed for the original $\varepsilon-$insensitive loss function.

In each iteration $k > 0$ of the fast gradient algorithm (cf. the algorithmic scheme in Section 4.4) that solves the dual problem ($\mathrm{D}_{\mu,\gamma}$) all we need to calculate is the gradient of $F_{\mu,\gamma}$ at point $w^k \in \mathbb{R}^n \times \ldots \times \mathbb{R}^n$ and the Lipschitz constant $L(\mu, \gamma)$. The latter we will take care of later. As seen in the theoretical considerations above the gradient is given by

$$\begin{aligned}\nabla F_{\mu,\gamma}(w^k) = &\left(K\sum_{i=1}^n w_i^k, \ldots, K\sum_{i=1}^n w_i^k\right) \\ &+ \left(\mathrm{Prox}_{\frac{1}{\mu_1}g_1}\left(\frac{w_1^k}{\mu_1}\right), \ldots, \mathrm{Prox}_{\frac{1}{\mu_n}g_n}\left(\frac{w_n^k}{\mu_n}\right)\right) + \gamma w^k. \qquad (5.34)\end{aligned}$$

The kernel matrix K is known and the iterates w^k are given in each iteration, i. e. all we need to calculate are the proximal points

$$\mathrm{Prox}_{\frac{1}{\mu_i}g_i}\left(\frac{w_i^k}{\mu_i}\right), \qquad \forall i = 1, \ldots, n,$$

the proximal point of parameter $\frac{1}{\mu_i}$ of g_i at point $\frac{w_i^k}{\mu_i}$. We calculate first this proximal point for arbitrary $z \in \mathbb{R}^n$ which is determined by solving the problem

$$\mathrm{Prox}_{\frac{1}{\mu_i}g_i}(z) = \operatorname*{argmin}_{x\in\mathbb{R}^n}\left\{g_i(x) + \frac{\mu_i}{2}\|z - x\|^2\right\},$$

and obtain by plugging in the definition of function g_i,

$$\begin{aligned}&= \operatorname*{argmin}_{x\in\mathbb{R}^n}\left\{C|\langle e_i, x\rangle - y_i|_\varepsilon + \delta_\Lambda(x) + \frac{\mu_i}{2}\|z - x\|^2\right\} \\ &= \operatorname*{argmin}_{x\in\Lambda}\left\{C|\langle e_i, x\rangle - y_i|_\varepsilon + \frac{\mu_i}{2}\sum_{i=1}^n (z_i - x_i)^2\right\} \\ &= \operatorname*{argmin}_{x\in\Lambda}\left\{C|x_i - y_i|_\varepsilon + \frac{\mu_i}{2}(z_i - x_i)^2 + \frac{\mu_i}{2}\sum_{\substack{j=1\\ j\neq i}}^n (z_j - x_j)^2\right\}.\end{aligned}$$

The above optimization problem can be solved by considering n separate minimization problems, each to be solved for one coordinate of the proximal point. Thus we get for $t = 1, \ldots, n$

$$\Big(\mathrm{Prox}_{\frac{1}{\mu_i} g_i}(z)\Big)_t = \operatorname*{argmin}_{\alpha \in [y_i - \xi, y_i + \xi]} \left\{ C|\alpha - y_i|_\varepsilon + \frac{\mu_i}{2}(z_i - \alpha)^2 \right\}, \qquad \text{if} \quad t = i,$$

and

$$\Big(\mathrm{Prox}_{\frac{1}{\mu_i} g_i}(z)\Big)_t = \operatorname*{argmin}_{\alpha \in [y_t - \xi, y_t + \xi]} \left\{ \frac{\mu_i}{2}(z_t - \alpha)^2 \right\}, \qquad \text{if} \quad t \neq i. \tag{5.35}$$

We first consider the case where $t = i$. Therefore, we introduce the function $h : \mathbb{R} \to \mathbb{R}$,

$$h(\alpha) = C|\alpha - y_i|_\varepsilon + \frac{\mu_i}{2}(z_i - \alpha)^2,$$

and consider the problem

$$\inf_{\alpha \in \mathbb{R}} \{h(\alpha)\}. \tag{5.36}$$

The point $\alpha^* \in \mathbb{R}$ is the unique minimizer of problem (5.36) if and only if

$$\begin{aligned} 0 \in \partial h(\alpha^*) &= \partial\Big(C|\cdot - y_i|_\varepsilon + \frac{\mu_i}{2}(z_i - \cdot)^2\Big)(\alpha^*) \\ &= C\partial\big(|\cdot - y_i|_\varepsilon\big)(\alpha^*) - \mu_i z_i + \mu_i \alpha^*. \end{aligned} \tag{5.37}$$

Since it holds

$$|a - y_i|_\varepsilon = \begin{cases} -a + y_i - \varepsilon, & \text{if } a < y_i - \varepsilon, \\ 0, & \text{if } y_i - \varepsilon \leq a \leq y_i + \varepsilon, \\ a - y_i - \varepsilon, & \text{if } a > y_i + \varepsilon, \end{cases}$$

we obtain for the subdifferential term in (5.37)

$$C\partial(|\cdot - y_i|_\varepsilon)(\alpha^*) = \begin{cases} -C, & \text{if } \alpha^* < y_i - \varepsilon, \\ [-C, 0], & \text{if } \alpha^* = y_i - \varepsilon, \\ 0, & \text{if } y_i - \varepsilon < \alpha^* < y_i + \varepsilon, \\ [0, C], & \text{if } \alpha^* = y_i + \varepsilon, \\ C, & \text{if } \alpha^* > y_i + \varepsilon. \end{cases} \tag{5.38}$$

Moreover, (5.37) is equivalent to

$$\mu_i z_i \in C\partial(|\cdot - y_i|_\varepsilon)(\alpha^*) + \mu_i \alpha^*.$$

Next we will have a look at each of the five cases in (5.38) separately by making use of the above inclusion.

(i) Taking into account the first case in (5.38) we get that if $\alpha^* < y_i - \varepsilon$ it holds $\mu_i z_i = -C + \mu_i \alpha^*$ which yields that

$$\alpha^* = \frac{\mu_i z_i + C}{\mu_i} \quad \text{if} \quad \frac{\mu_i z_i + C}{\mu_i} < y_i - \varepsilon,$$

i. e. if $\mu_i z_i < \mu_i(y_i - \varepsilon) - C$.

(ii) Taking into account the second case in (5.38) we get that if $\alpha^* = y_i - \varepsilon$ it holds $\mu_i z_i \in [-C + \mu_i \alpha^*, \mu_i \alpha^*]$ or, equivalently, $\mu_i(y_i - \varepsilon) - C \leq \mu_i z_i \leq \mu_i(y_i - \varepsilon)$.

(iii) Taking into account the third case in (5.38) we get that if $y_i - \varepsilon < \alpha^* < y_i + \varepsilon$ it holds that $\mu_i z_i = \mu_i \alpha^*$ or, equivalently, $\alpha^* = z_i$ if $\mu_i(y_i - \varepsilon) < \mu_i z_i < \mu_i(y_i + \varepsilon)$.

(iv) Taking into account the fourth case in (5.38) we get that if $\alpha^* = y_i + \varepsilon$ it holds that $\mu_i z_i \in [\mu_i \alpha^*, C + \mu_i \alpha^*]$ or, equivalently, $\mu_i(y_i + \varepsilon) \leq \mu_i z_i \leq \mu_i(y_i + \varepsilon) + C$.

(v) Taking into account the fifth case in (5.38) we get that if $\alpha^* > y_i + \varepsilon$ it holds $\mu_i z_i = C + \mu_i \alpha^*$ which yields that

$$\alpha^* = \frac{\mu_i z_i - C}{\mu_i} \quad \text{if} \quad \frac{\mu_i z_i - C}{\mu_i} > y_i + \varepsilon,$$

i. e. if $\mu_i z_i > \mu_i(y_i + \varepsilon) + C$.

So far we derived the optimal solution α^* to problem (5.36) where the observations $(i) - (v)$ can be summarized by

$$\alpha^* = \begin{cases} \frac{\mu_i z_i + C}{\mu_i}, & \text{if } \mu_i z_i < \mu_i(y_i - \varepsilon) - C, \\ y_i - \varepsilon, & \text{if } \mu_i(y_i - \varepsilon) - C \leq \mu_i z_i \leq \mu_i(y_i - \varepsilon), \\ z_i, & \text{if } \mu_i(y_i - \varepsilon) < \mu_i z_i < \mu_i(y_i + \varepsilon), \\ y_i + \varepsilon, & \text{if } \mu_i(y_i + \varepsilon) \leq \mu_i z_i \leq \mu_i(y_i + \varepsilon) + C, \\ \frac{\mu_i z_i - C}{\mu_i}, & \text{if } \mu_i z_i > \mu_i(y_i + \varepsilon) + C. \end{cases} \tag{5.39}$$

Finally, to obtain the $i-$th component of the proximal point $\operatorname{Prox}_{\frac{1}{\mu_i} g_i}(z)$ it remains to project α^* onto the boundaries of the feasible interval if $\alpha^* \notin [y_i - \xi, y_i + \xi]$. Thus, we obtain

$$\left(\operatorname{Prox}_{\frac{1}{\mu_i} g_i}(z)\right)_i = \operatorname{Proj}_{[y_i - \xi, y_i + \xi]}(\alpha^*). \tag{5.40}$$

For all other components $t \neq i$ of the proximal point we consider (5.35). Introducing the minimization problem

$$\inf_{\alpha \in \mathbb{R}} \left\{ \frac{\mu_i}{2}(z_t - \alpha)^2 \right\}$$

with differentiable objective function we derive the minimizer to be equal to $\alpha^* = z_t$, $t = 1, \ldots, n$, $t \neq i$. Thus we have

$$\left(\mathrm{Prox}_{\frac{1}{\mu_i} g_i}(z)\right)_t = \mathrm{Proj}_{[y_t - \xi, y_t + \xi]}(z_t), \qquad \forall t \neq i. \tag{5.41}$$

Since we are interested in each iteration $k \geq 0$, for all $i = 1, \ldots, n$, in the proximal point

$$\mathrm{Prox}_{\frac{1}{\mu_i} g_i}\left(\frac{w_i^k}{\mu_i}\right) \tag{5.42}$$

where w_i^k is the $i-$th component of the iterates $w^k = (w_1^k, \ldots, w_n^k) \in \mathbb{R}^n \times \ldots \times \mathbb{R}^n$ in the fast gradient scheme, we simply have to set $z = \frac{w_i^k}{\mu_i}$ and obtain in analogy to (5.39)

$$\alpha^* = \begin{cases} \frac{(w_i^k)_i + C}{\mu_i}, & \text{if } (w_i^k)_i < \mu_i(y_i - \varepsilon) - C, \\ y_i - \varepsilon, & \text{if } \mu_i(y_i - \varepsilon) - C \leq (w_i^k)_i \leq \mu_i(y_i - \varepsilon), \\ \frac{(w_i^k)_i}{\mu_i}, & \text{if } \mu_i(y_i - \varepsilon) < (w_i^k)_i < \mu_i(y_i + \varepsilon), \\ y_i + \varepsilon, & \text{if } \mu_i(y_i + \varepsilon) \leq (w_i^k)_i \leq \mu_i(y_i + \varepsilon) + C, \\ \frac{(w_i^k)_i - C}{\mu_i}, & \text{if } (w_i^k)_i > \mu_i(y_i + \varepsilon) + C \end{cases}$$

for the value to be projected on the interval $[y_i - \xi, y_i + \xi]$, i. e. the value which has to be considered in (5.40) to obtain the $i-$th component of the desired proximal point (5.42). Furthermore, the remaining components of the proximal point of function g_i are given by

$$\left(\mathrm{Prox}_{\frac{1}{\mu_i} g_i}\left(\frac{w_i^k}{\mu_i}\right)\right)_t = \mathrm{Proj}_{[y_t - \xi, y_t + \xi]}\left(\frac{(w_i^k)_t}{\mu_i}\right), \qquad \forall t = 1, \ldots, n, \quad t \neq i,$$

$i = 1, \ldots, n$.

The quadratic $\varepsilon-$insensitive loss function

Recall that the quadratic $\varepsilon-$insensitive loss function $v_{\varepsilon^2} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is defined by

$$v_{\varepsilon^2}(a, y) = (|a - y|_\varepsilon)^2 = \begin{cases} 0, & \text{if } |a - y| \leq \varepsilon, \\ (|a - y| - \varepsilon)^2, & \text{else.} \end{cases}$$

With the same argumentation as in the case of the $\varepsilon-$insensitive loss function above we get for our primal problem ($\mathrm{P}_{\mathrm{gen}}$) the functions

$$g_i(\cdot) = C(|\langle e_i, \cdot \rangle - y|_\varepsilon)^2 + \delta_\Lambda(\cdot).$$

To be able to use the fast gradient algorithm efficiently we again have to provide the proximal points occurring in the gradient (5.34). The formula for the proximal points for $i = 1, \ldots, n$ now becomes

$$\begin{aligned}
\mathrm{Prox}_{\frac{1}{\mu_i} g_i}(z) &= \operatorname*{argmin}_{x \in \mathbb{R}^n} \left\{ g_i(x) + \frac{\mu_i}{2} \|z - x\|^2 \right\} \\
&= \operatorname*{argmin}_{x \in \mathbb{R}^n} \left\{ C(|\langle e_i, x \rangle - y_i|_\varepsilon)^2 + \delta_\Lambda(x) + \frac{\mu_i}{2} \|z - x\|^2 \right\} \\
&= \operatorname*{argmin}_{x \in \Lambda} \left\{ C(|x_i - y_i|_\varepsilon)^2 + \frac{\mu_i}{2}(z_i - x_i)^2 + \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\mu_i}{2} (z_j - x_j)^2 \right\}.
\end{aligned}$$

We consider now n different optimization problems separately, one for each coordinate of the corresponding proximal point. Thus, for $t = 1, \ldots, n$ we have

$$\left(\mathrm{Prox}_{\frac{1}{\mu_i} g_i}(z)\right)_t = \operatorname*{argmin}_{\alpha \in [y_i - \xi, y_i + \xi]} \left\{ C(|\alpha - y_i|_\varepsilon)^2 + \frac{\mu_i}{2}(z_i - \alpha)^2 \right\}, \qquad \text{if} \quad t = i, \tag{5.43}$$

and

$$\left(\mathrm{Prox}_{\frac{1}{\mu_i} g_i}(z)\right)_t = \operatorname*{argmin}_{\alpha \in [y_t - \xi, y_t + \xi]} \left\{ \frac{\mu_i}{2}(z_t - \alpha)^2 \right\}, \qquad \text{if} \quad t \neq i.$$

Notice that the latter problems for $t = 1, \ldots, n$, $t \neq i$, are identical to the problems formulated in (5.35). So the work on these components of the proximal points has been done in the previous calculations and we can concentrate on the problem (5.43). Therefore, we introduce the function $h : \mathbb{R} \to \mathbb{R}$,

$$h(\alpha) = C(|\alpha - y|_\varepsilon)^2 + \frac{\mu_i}{2}(z_i - \alpha)^2.$$

and will take care for the problem

$$\inf_{\alpha \in \mathbb{R}} h(\alpha).$$

Notice that we can write

$$(|\alpha - y|_\varepsilon)^2 = \begin{cases} (-\alpha + y_i - \varepsilon)^2, & \text{if } \alpha < y_i - \varepsilon, \\ 0, & \text{if } y_i - \varepsilon \leq \alpha \leq y_i + \varepsilon, \\ (\alpha - y_i - \varepsilon)^2, & \text{if } \alpha > y_i + \varepsilon. \end{cases}$$

and we obtain, unlike in the case of the $\varepsilon-$insensitive loss function treated above, a continuously differentiable objective function h. The derivative h is then given by

$$\frac{\mathrm{d}h(\alpha)}{\mathrm{d}\alpha} = \begin{cases} (2C+\mu_i)\alpha - 2C(y_i-\varepsilon) - \mu_i z_i, & \text{if } \alpha < y_i - \varepsilon, \\ -\mu_i z_i + \mu_i \alpha, & \text{if } y_i - \varepsilon \leq \alpha \leq y_i + \varepsilon, \\ (2C+\mu_i)\alpha - 2C(y_i+\varepsilon) - \mu_i z_i, & \text{if } \alpha > y_i + \varepsilon. \end{cases} \tag{5.44}$$

Using this formula we can calculate minimizer of $\inf_{\alpha\in\mathbb{R}}\{h(\alpha)\}$. Considering the first case in (5.44), we obtain that if $\alpha < y_i - \varepsilon$ then the minimizer α^* fulfills

$$(2C+\mu_i)\alpha^* - 2C(y_i-\varepsilon) - \mu_i z_i = 0, \quad \text{i. e.} \quad \alpha^* = \frac{2C(y_i-\varepsilon)+\mu_i z_i}{2C+\mu_i}.$$

With that, the condition $\alpha < y_i - \varepsilon$ then becomes

$$\frac{2C(y_i-\varepsilon)+\mu_i z_i}{2C+\mu_i} < y_i - \varepsilon, \quad \text{i. e.} \quad \mu_i z_i < \mu_i(y_i-\varepsilon).$$

For the second case in (5.44) we obtain that $\alpha^* = z_i$ if $\mu_i(y_i-\varepsilon) \leq \mu_i z_i \leq \mu_i(y_i+\varepsilon)$. Finally, for the third case in (5.44) we get analogously to the first,

$$\alpha^* = \frac{2C(y_i+\varepsilon)+\mu_i z_i}{2C+\mu_i} \quad \text{if} \quad \mu_i z_i > \mu_i(y_i+\varepsilon).$$

Summarizing these observations the unique minimizer α^* of $\inf_{\alpha\in\mathbb{R}}\{h(\alpha)\}$ is given by

$$\alpha^* = \begin{cases} \frac{2C(y_i-\varepsilon)+\mu_i z_i}{2C+\mu_i}, & \text{if } \mu_i z_i < \mu_i(y_i-\varepsilon), \\ z_i, & \text{if } \mu_i(y_i-\varepsilon) \leq \mu_i z_i \leq \mu_i(y_i+\varepsilon), \\ \frac{2C(y_i+\varepsilon)+\mu_i z_i}{2C+\mu_i}, & \text{if } \mu_i z_i > \mu_i(y_i+\varepsilon). \end{cases}$$

Again, we are interested in the $i-$th coordinate of the proximal points of g_i of parameter $\frac{1}{\mu_i}$ at $\frac{w_i^k}{\mu_i}$, for $i = 1, \ldots, n$. Thus, by setting $z_i = \frac{(w_i^k)_i}{\mu_i}$ we get for the minimizer

$$\alpha^* = \begin{cases} \frac{2C(y_i-\varepsilon)+(w_i^k)_i}{2C+\mu_i}, & \text{if } (w_i^k)_i < \mu_i(y_i-\varepsilon), \\ z_i, & \text{if } \mu_i(y_i-\varepsilon) \leq (w_i^k)_i \leq \mu_i(y_i+\varepsilon), \\ \frac{2C(y_i+\varepsilon)+(w_i^k)_i}{2C+\mu_i}, & \text{if } (w_i^k)_i > \mu_i(y_i+\varepsilon). \end{cases} \tag{5.45}$$

by projecting this value onto the interval $[y_i - \xi, y_i + \xi]$ we obtain the desired proximal points, i. e.

$$\left(\mathrm{Prox}_{\frac{1}{\mu_i}g_i}\left(\frac{w_i^k}{\mu_i}\right)\right)_i = \mathrm{Proj}_{[y_i-\xi, y_i+\xi]}\left(\alpha^*\right),$$

with α^* given by (5.45).

The Huber loss function

In the following we will consider the Huber loss function

$$v_{\mathrm{H}}(a,y) = \begin{cases} \varepsilon|a-y| - \frac{\varepsilon^2}{2}, & \text{if } |a-y| > \varepsilon, \\ \frac{1}{2}|a-y|^2, & \text{if } |a-y| \leq \varepsilon \end{cases}$$

$$= \begin{cases} -\varepsilon(a-y) - \frac{\varepsilon^2}{2}, & \text{if } a < y-\varepsilon, \\ \frac{1}{2}(a-y)^2, & \text{if } y-\varepsilon \leq a \leq y+\varepsilon, \\ \varepsilon(a-y) - \frac{\varepsilon^2}{2}, & \text{if } a > y+\varepsilon, \end{cases}$$

where it is worth noticing that in that case $\varepsilon > 0$ does not correspond to the width of the tube around the input data. Instead it corresponds to the distance from the input value where the type of penalization is changed. The functions $g_i : \mathbb{R}^n \to \overline{\mathbb{R}}$ for the primal problem $(\mathrm{P}_{\mathrm{gen}})$ now are

$$g_i(\cdot) = Cv_i(\cdot) + \delta_\Lambda(\cdot),$$

$i = 1, \ldots, n$, where $v_i : \mathbb{R}^n \to \mathbb{R}$, $v_i(x) = v_{\mathrm{H}}(x_i, y_i)$. Again, like in the previous considerations for the $\varepsilon-$insensitive loss function and the quadratic $\varepsilon-$insensitive loss function we need to calculate the corresponding proximal points for the gradient (5.34). Thus, we need to find the unique minimizer of

$$\mathrm{Prox}_{\frac{1}{\mu_i}g_i}(z) = \operatorname*{argmin}_{x\in\mathbb{R}^n} \left\{ Cv_i(x) + \delta_\Lambda(x) + \frac{\mu_i}{2}\|z-x\|^2 \right\}$$

$$= \operatorname*{argmin}_{x\in\Lambda} \left\{ Cv_{\mathrm{H}}(x_i,y_i) + \frac{\mu_i}{2}(z_i-x_i)^2 + \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{\mu_i}{2}(z_j-x_j)^2 \right\}.$$

Here we can consider n optimization problems separately, each for one component for the proximal points of g_i of parameter $\frac{1}{\mu_i}$ at z, $i = 1, \ldots, n$. Again we only have to account for the problem whose minimizer is the $i-$th component since the remaining part has already been done, i. e. consider

$$\left(\mathrm{Prox}_{\frac{1}{\mu_i}g_i}(z)\right)_i = \operatorname*{argmin}_{x_i\in[y_i-\xi,y_i+\xi]} \left\{ Cv_{\mathrm{H}}(x_i,y_i) + \frac{\mu_i}{2}(z_i-x_i)^2 \right\}.$$

We define the function $h : \mathbb{R} \to \mathbb{R}$,

$$h(\alpha) = Cv_{\mathrm{H}}(\alpha, y_i) + \frac{\mu_i}{2}(z_i-\alpha)^2$$

and solve the problem $\inf_{\alpha\in\mathbb{R}}\{h(\alpha)\}$. Since v_{H} is differentiable w. r. t. its first component and the derivative is given by

$$v'_{\mathrm{H}}(a, y_i) = \begin{cases} -\varepsilon, & \text{if } a < y_i - \varepsilon, \\ a - y_i, & \text{if } y_i - \varepsilon \leq a \leq y_i + \varepsilon, \\ \varepsilon, & \text{if } a > y_i + \varepsilon, \end{cases} \tag{5.46}$$

we obtain three cases. Taking into account the first case in (5.46) we get that α^* is a minimizer of $\inf_{\alpha\in\mathbb{R}}\{h(\alpha)\}$ if

$$C(-\varepsilon) - \mu_i z_i + \mu_i \alpha^* = 0, \quad \text{i. e.} \quad \alpha^* = \frac{C\varepsilon + \mu_i z_i}{\mu_i}$$

if $\mu_i z_i < \mu_i(y_i - \varepsilon) - C\varepsilon$. For the second case we obtain that

$$\alpha^* = \frac{C y_i + \mu_i z_i}{C + \mu_i}$$

is a minimizer if $\mu_i(y_i - \varepsilon) - C\varepsilon \leq \mu_i z_i \leq \mu_i(y_i + \varepsilon) + C\varepsilon$. Finally, the last case yields

$$\alpha^* = \frac{-C\varepsilon + \mu_i z_i}{\mu_i} \quad \text{if} \quad \mu_i z_i > \mu_i(y_i + \varepsilon) + C\varepsilon.$$

Since we need in each iteration $k \geq 0$ of the fast gradient algorithm the corresponding proximal point at $\frac{w_i^k}{\mu_i}$ we set $z_i = \frac{(w_i^k)_i}{\mu_i}$ and obtain

$$\alpha^* = \begin{cases} \frac{C\varepsilon + (w_i^k)_i}{\mu_i}, & \text{if } (w_i^k)_i < \mu_i(y_i - \varepsilon) - C\varepsilon, \\ \frac{C y_i + (w_i^k)_i}{C + \mu_i}, & \text{if } \mu_i(y_i - \varepsilon) - C\varepsilon \leq (w_i^k)_i \leq \mu_i(y_i + \varepsilon) + C\varepsilon, \\ \frac{-C\varepsilon + (w_i^k)_i}{\mu_i}, & \text{if } (w_i^k)_i > \mu_i(y_i + \varepsilon) + C\varepsilon, \end{cases}$$

that in a last step has to be projected onto the interval $[y_i - \xi, y_i + \xi]$, i. e.

$$\left(\mathrm{Prox}_{\frac{1}{\mu_i} g_i}(z)\right)_i = \mathrm{Proj}_{[y_i - \xi, y_i + \xi]}(\alpha^*).$$

The extended loss function

Now we will have a look at the last loss function for regression we will consider in this thesis. Recall the extended loss function $v_{\mathrm{ext}} : \mathbb{R} \times \mathbb{R} \to \overline{\mathbb{R}}$,

$$v_{\mathrm{ext}}(a, y) = \delta_{[-\varepsilon,\varepsilon]}(a - y) = \begin{cases} 0, & \text{if } |a - y| \leq \varepsilon, \\ +\infty, & \text{else.} \end{cases}$$

The functions g_i, $i = 1 \ldots, n$, for the primal problem ($\mathrm{P}_{\mathrm{gen}}$) are then defined as $g_i(\cdot) = Cv_i(\cdot)$, where $v_i : \mathbb{R}^n \to \overline{\mathbb{R}}$, $v_i(x) = v_{\mathrm{ext}}(x_i, y_i)$. Notice that in the case of the extended loss function as considered here we do not use the set Λ to get a bounded domain of the functions g_i. Instead, the set $\Lambda_\varepsilon := [y_1 - \varepsilon, y_1 + \varepsilon] \times \ldots \times [y_n - \varepsilon, y_n + \varepsilon]$ arises as a natural choice to get the domains bounded. The corresponding proximal points of g_i of parameter $\frac{1}{\mu_i}$ at $z \in \mathbb{R}^n$, $i = 1, \ldots, n$, are then

$$\begin{aligned}
\mathrm{Prox}_{\frac{1}{\mu_i} g_i}(z) &= \operatorname*{argmin}_{x \in \mathbb{R}^n} \left\{ Cv_i(x) + \delta_{\Lambda_\varepsilon}(x) + \frac{\mu_i}{2} \|z - x\|^2 \right\} \\
&= \operatorname*{argmin}_{x \in \Lambda_\varepsilon} \left\{ C\delta_{[-\varepsilon,\varepsilon]}(x_i - y_i) + \frac{\mu_i}{2} \|z - x\|^2 \right\} \\
&= \operatorname*{argmin}_{x \in \Lambda_\varepsilon} \left\{ C\delta_{[-\varepsilon,\varepsilon]}(x_i - y_i) + \frac{\mu_i}{2}(z_i - x_i)^2 + \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{\mu_i}{2}(z_j - x_j)^2 \right\}.
\end{aligned}$$

Again we solve n separate minimization problems to obtain the proximal point for each of the functions g_i, $i = 1, \ldots, n$, i. e. we calculate each component $t = 1, \ldots, n$ for the proximal point of g_i. For the component $t = i$ we have to consider

$$\begin{aligned}
\left(\mathrm{Prox}_{\frac{1}{\mu_i} g_i}(z) \right)_i &= \operatorname*{argmin}_{x_i \in [y_i - \varepsilon, y_i + \varepsilon]} \left\{ C\delta_{[-\varepsilon,\varepsilon]}(x_i - y_i) + \frac{\mu_i}{2}(z_i - x_i)^2 \right\} \\
&= \operatorname*{argmin}_{x_i \in [y_i - \varepsilon, y_i + \varepsilon]} \left\{ C\delta_{[y_i - \varepsilon, y_i + \varepsilon]}(x_i) + \frac{\mu_i}{2}(z_i - x_i)^2 \right\} \\
&= \operatorname*{argmin}_{x_i \in [y_i - \varepsilon, y_i + \varepsilon]} \left\{ \frac{\mu_i}{2}(z_i - x_i)^2 \right\}
\end{aligned}$$

which turns out to be the same as for all other components $t \neq i$. The unique minimizer of this problem is given by the projection of the minimizer α^* of

$$\inf_{\alpha \in \mathbb{R}} \left\{ \frac{\mu_i}{2}(z_i - \alpha)^2 \right\}$$

to the corresponding interval, which is $\alpha^* = z_i$. As we need again the proximal point at $z = \frac{w_i^k}{\mu_i}$ we set $z_t = \frac{(w_i^k)_t}{\mu_i}$ and finally obtain

$$\left(\mathrm{Prox}_{\frac{1}{\mu_i} g_i} \left(\frac{w_i^k}{\mu_i} \right) \right)_t = \mathrm{Proj}_{[y_t - \varepsilon, y_t + \varepsilon]} \left(\frac{(w_i^k)_t}{\mu_i} \right)$$

for the $t-$th component, $t = 1, \ldots, n$, of the proximal point of g_i, $i = 1, \ldots, n$.

5.2.3 SVR Numerical Results

We will use here the same setting of the regression task like in Subsection 3.5.1 with the toy data set as input data. But the important difference is that we do not need to reformulate the dual optimization problem to get a formulation that can be solved via standard solver as done before. We will numerically solve the dual problem ($\mathrm{D}_{\mathrm{gen}}$) directly within $\varepsilon-$accuracy by solving the doubly smoothed problem ($\mathrm{D}_{\mu,\gamma}$) via the fast gradient scheme.

To be able to apply the fast gradient algorithm all we need is the gradient of the doubly smoothed function $F_{\mu,\gamma}$ (cf. (5.34)) and the Lipschitz constant of the gradient. The proximal points have been calculated in Subsection 5.2.2 for the four loss functions effecting the appearance of the functions g_i, $i = 1, \ldots, n$, in the dual problem ($\mathrm{D}_{\mathrm{gen}}$). Therefore, it remains to calculate the Lipschitz constant in each case, i. e. for each optimization problem arising with the use of different loss functions. We have seen in Subsection 5.2.1 the Lipschitz constant of the gradient of $F_{\mu,\gamma}$ is

$$L(\mu, \gamma) = \max_{i=1,\ldots,n} \left\{ \frac{1}{\mu_i} \right\} + n\|K\| + \gamma. \tag{5.47}$$

First, the first summand in the formula for $L(\mu, \gamma)$ will be treated. For each function g_i^* we have chosen the smoothing parameter μ_i, $i = 1, \ldots, n$, in order to obtain a smooth approximation g_{i,μ_i}^* of it. In Subsection 5.2.1 we have seen that, in order to obtain $\varepsilon-$accuracy, these smoothing parameters have to be set to

$$\mu_i = \frac{\varepsilon}{(n+2)D_{g_i}},$$

(see (5.25)). That means

$$\max_{i=1,\ldots,n} \left\{ \frac{1}{\mu_i} \right\} = \max_{i=1,\ldots,n} \left\{ \frac{(n+2)D_{g_i}}{\varepsilon} \right\} = \frac{(n+2)}{\varepsilon} \max_{i=1,\ldots,n} \{D_{g_i}\}$$

and to determine these smoothing parameters we will need the values D_{g_i} which were defined to be

$$D_{g_i} = \sup_{x \in \mathrm{dom}\, g_i} \left\{ \frac{1}{2}\|x\|^2 \right\}.$$

In the case of using the $\varepsilon-$insensitive, the quadratic $\varepsilon-$insensitive and the Huber loss function the functions g_i are of type

$$g_i(\cdot) = Cv_i(\cdot) + \delta_\Lambda(\cdot), \tag{5.48}$$

where $\Lambda = [y_1 - \xi, y_1 + \xi] \times \ldots \times [y_n - \xi, y_n + \xi]$ and the functions v_i are defined based on the corresponding loss functions used. In all three cases, however,

the domain of all functions g_i, is the equal to the set Λ, i. e. $\operatorname{dom} g_i = \Lambda$ for all $i = 1, \ldots, n$. Notice that here, unlike in the general case, $\operatorname{dom} g_i \subset \mathbb{R}^n$ instead of $\operatorname{dom} g_i \subset \mathbb{R}^{k_i}$ since the operators K_i are all identical to $K : \mathbb{R}^n \to \mathbb{R}^n$ corresponding to the chosen kernel. Thus,

$$D_{g_i} = \sup_{x \in \operatorname{dom} g_i} \left\{ \frac{1}{2}\|x\|^2 \right\} = \sup_{x \in \Lambda} \left\{ \frac{1}{2} \sum_{i=1}^{n} x_i^2 \right\} = \sum_{i=1}^{n} \sup_{x_i \in [y_i - \xi, y_i + \xi]} \left\{ \frac{1}{2} x_i^2 \right\}.$$

For all $i = 1, \ldots, n$ we obtain

$$\sup_{x_i \in [y_i - \xi, y_i + \xi]} \left\{ \frac{1}{2} x_i^2 \right\} = \max \left\{ \frac{1}{2}(y_i - \xi)^2, \frac{1}{2}(y_i + \xi)^2 \right\}$$

and, since the input points y_i are given and ξ has to be determined in advance, D_{g_i} can be easily computed. In that special case where the domains $\operatorname{dom} g_i$ are all identical it consequently suffices to calculate it once to find the maximum among all D_{g_i}.

We turn now to the case when the extended loss function is applied. Here we modified the last term in (5.48) to be $\delta_{\Lambda_\varepsilon}(\cdot)$ which corresponds of setting $\xi = \varepsilon$ in the above considerations and the values D_{g_i} are found as easy as in the case of the other three loss functions.

With that the first term for the formula of the Lipschitz constant (5.47) is checked. In the second term, namely $n\|K\|$ the norm of the kernel matrix has to be computed. Last, the smoothing parameter γ is obtained by taking into account that

$$\gamma = \frac{2\varepsilon}{(n+2)R^2},$$

(see (5.25)) where R is an upper bound for the norm of the optimal solution of the dual problem ($\mathrm{D}_{\mathrm{gen}}$), i. e. $\|p^*\|_{\bar{n}} \leq R$.

Again we will sample the input data from the function $\operatorname{sinc} : \mathbb{R} \to \mathbb{R}$,

$$\operatorname{sinc}(x) = \begin{cases} \frac{\sin x}{x}, & \text{if } x \neq 0, \\ 1, & \text{else,} \end{cases}$$

where we take the function values at points $X = \{-5.0, -4.8, \ldots, 4.8, 5.0\} \subset \mathbb{R}$ resulting in $n = 51$ input samples $(x_i, \operatorname{sinc}(x_i))_{i=1}^n \subset X \times \mathbb{R}$ that are perturbed by adding a zero mean Gaussian noise with variance $\sigma^2 = 0.05$. Thus, we obtain the training set $\mathcal{D} = \{(x_i, y_i) : i = 1, \ldots, n\}$, where the y_i are the perturbed values $\operatorname{sinc}(x_i)$, $i = 1, \ldots, n$. The shape of the original function together with the sampled points is shown in Figure 5.11. Like in Subsection 3.5.1 we use

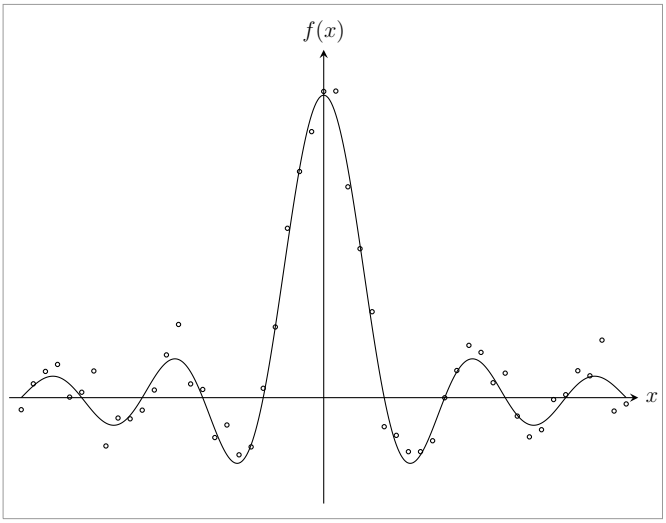


Figure 5.11: Plot of the sinc function and the corresponding training samples.

the Gaussian kernel function (cf. (3.15)) and $K \in \mathbb{R}^{n \times n}$ denotes the symmetric positive definite kernel matrix.

Our first example will be the application of the $\varepsilon-$insensitive loss function. We will denote in the following by $\varepsilon_{\mathrm{DS}} > 0$ the epsilon value responsible for the accuracy of the double smoothing algorithm and by $\varepsilon > 0$ the epsilon value occurring in the corresponding loss functions. As we use the Gaussian kernel function we set the kernel parameter $\sigma = 0.5$. The regularization parameter C is set to be equal to 1. Furthermore, we set $\varepsilon = 0.05$, $\xi = 0.15$. Concerning the parameters for the fast gradient algorithm we use $\varepsilon_{\mathrm{DS}} = 0.05$ and $R = 3.5$, where the latter has been found experimentally. The algorithm ran 20 000 iterations and we obtained the regression function by using the $\varepsilon-$insensitive loss function shown in Figure 5.12. In Figure 5.16(a) the behavior of the norm of the gradient of the function $F_\mu(p^k)$ for iterations 0 to 20 000 is shown. Next we turn to the case of the quadratic $\varepsilon-$insensitive loss function. Here all parameters concerning the support vector regression specific parameters as well as the parameters belonging to the double smoothing algorithm remained the same with one exception. The constant for the upper bound of the optimal solution p^* of $(\mathrm{D}_{\mathrm{gen}})$ was set to $R = 0.5$. Figure 5.13 shows the result for employing the quadratic $\varepsilon-$insensitive loss function while Figure 5.16(b) shows the decrease of the norm of the gradient in this case. As a third example, let us employ the Huber loss function. Here, all parameters despite of the regularization parameter C are the same as in the previous example when using the quadratic $\varepsilon-$insensitive loss function. Here $C = 10$ was applied. In Figure 5.14 the resulting regression function is shown. Notice that we did not plot the $\varepsilon-$tube since in the case of the Huber loss

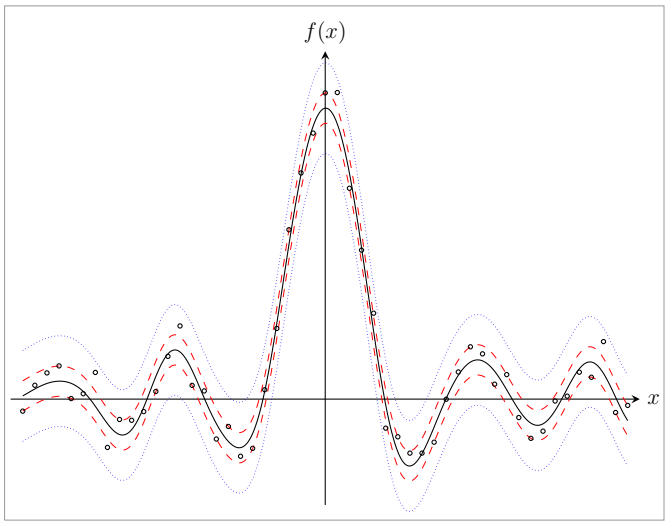


Figure 5.12: The resulting regression function (solid black line) for the $\varepsilon-$insensitive loss function together with the $\varepsilon-$tube (dashed red lines) and the $\xi-$tube (densely dotted blue lines).

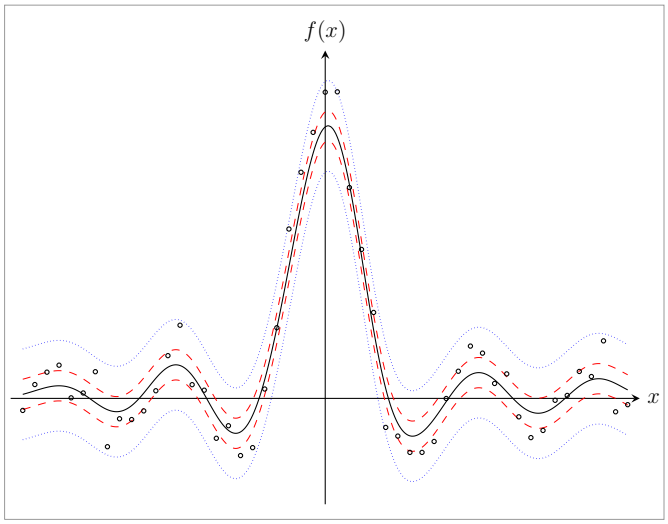


Figure 5.13: The resulting regression function (solid black line) for the quadratic $\varepsilon-$insensitive loss function together with the $\varepsilon-$tube (dashed red lines) and the $\xi-$tube (densely dotted blue lines).

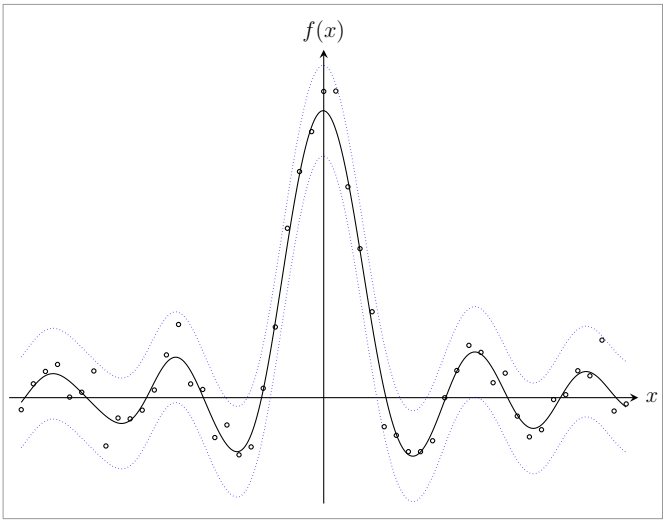


Figure 5.14: The resulting regression function (solid black line) for the Huber loss function together with the $\xi-$tube (densely dotted blue lines).

function this value determines the change of the type of penalization and not the value up to which deviations from the sampled value are not penalized like in the previous two cases. In Figure 5.16(c) the corresponding decrease of $\left\|\nabla F_\mu(p^k)\right\|_{\bar{n}}$ can be seen. In the last example we consider the extended loss function. Here we set $\varepsilon = \xi = 0.1$ and $R = 10$. The results can be seen in Figure 5.15 and Subfigure 5.16(d).

Remark 5.2. Notice that the numerical tests in this section are devoted only to illustrate that the support vector regression task can indeed be solved by the double smoothing technique. We do not apply here a cross validation analysis or other methods to determine optimal parameter combinations. Further, we do not measure the resulting accuracy of the regression as done in Subsection 3.5.1 since this is not the aim.

Remark 5.3. A Matlab implementation that solves the regression tasks considered in this Section can be found in the appendix. In particular, the implementations for the $\varepsilon-$insensitive loss function, the quadratic $\varepsilon-$insensitive loss function, the Huber loss function and the extended loss function are contained in subsections A.2.1, A.2.2, A.2.3 and A.2.4, respectively.

Remark 5.4. Numerical experiments have shown that the double smoothing approach performs bad on solving location problems. Primal-dual splitting algorithms like in [13] by far outperform our approach on such problems.

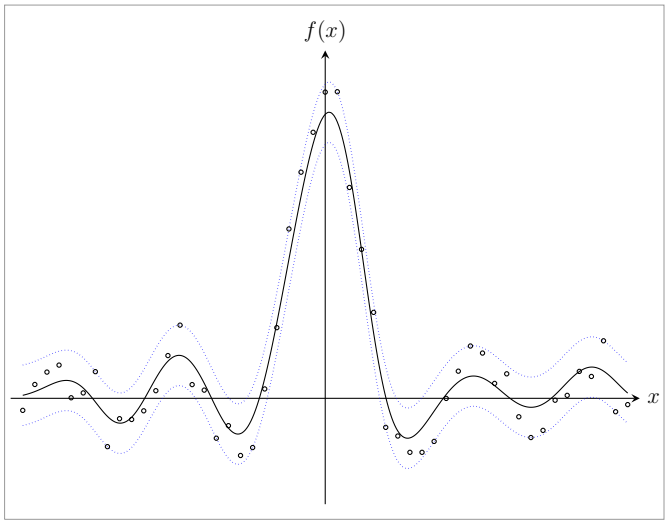
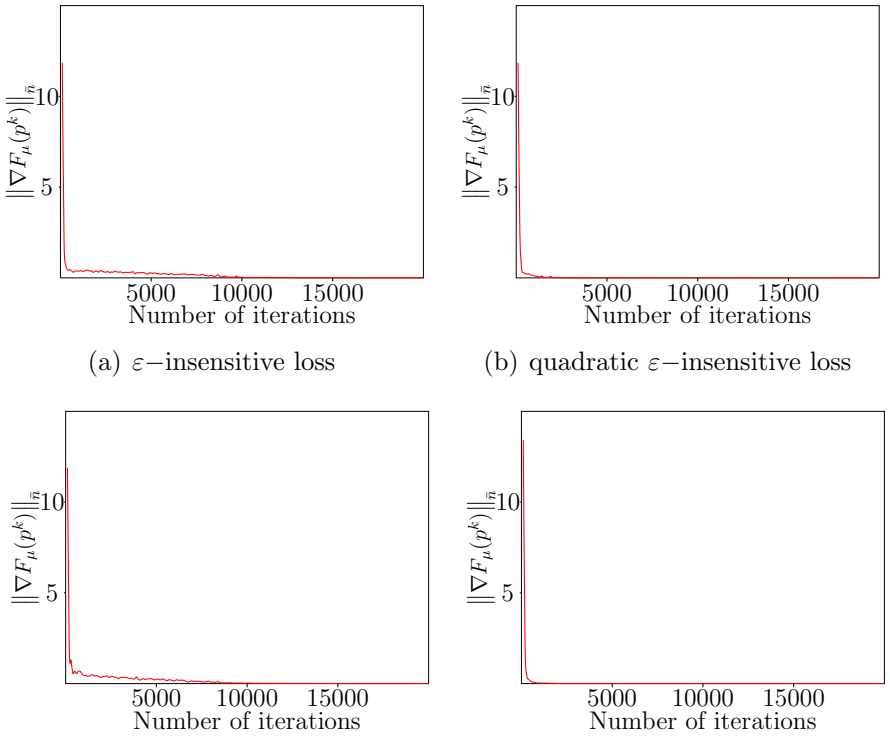


Figure 5.15: The resulting regression function (solid black line) for the extended loss function together with the $\xi-$tube (densely dotted blue lines).



(a) $\varepsilon-$insensitive loss

(b) quadratic $\varepsilon-$insensitive loss

(c) Huber loss

(d) extended loss

Figure 5.16: Illustrations of the decrease of the norm of the gradient of $F_\mu(p^k)$ dependent on the number of iterations.

Appendix A

Appendix

A.1 Imaging source code

A.1.1 Lena test image

Matlab source code for solving the image restoration task described in Subsection 5.1.2 for the Lena test image. The m-file containing this source code can be found on the compact disk attached to this thesis (*SourceCode/ImageRestoration/LenaImageRestoration.m*).

```
% initialize parameters
noiseFac    = 0.001; % factor for additive noise
R           = 0.05;  % upper bound for the norm of dual
                     % solution
paraLambda = 2e-6;   % regularization parameter
epsi        = 0.05;  % accuracy for the double smoothing
                     % algorithm

% specify the number of iterations
maxIterations = 100;

% read the image from the file, image file must lie in the
% same folder as this m-file
[pdata,~] = imread('lena.gif');

% show the original image
figure('Name','original image')
imshow(pdata)

% specify the lower and upper bound of the interval the
% pixels have to be scaled to
S = [0,0.1];
```

```
23
24 % convert and scale the original pixel data to [0,1]
25 X = double(pdata);
26 X = X./255;
27
28 % initialize the Gaussian lowpass filter of size 9 times 9
29 % and standard deviation 4
30 fsp = fspecial('gaussian',[9 9],4);
31
32 % blurr the scaled original image
33 B = imfilter(X,fsp,'conv','symmetric');
34
35 % initialize the noise matrix to be added to the blurred
36 % image B and add the noise
37 rng(57977,'v4');
38 noise = noiseFac*randn(size(X));
39
40 noisyB = B + noise;
41
42 % show the blurred and noisy image
43 figure('Name','blurred + noisy');
44 imshow(noisyB)
45
46 % scale the blurred and noisy image to the specified
47 % interval
48 noisyB = S(2).*noisyB;
49
50 % get the dimensions of the image
51 [n,m] = size(noisyB);
52
53 % specify the constants D_f and D_g
54 Df = 327.68;
55 Dg = 327.68;
56
57 % specify the smoothing parameters
58 paraRho   = epsi/(4*Df);
59 paraMu    = epsi/(4*Dg);
60 paraGamma = (2*epsi)/(4*R^2);
61
62 % specify the Lipschitz constant of the gradient
63 L = 1/paraMu + 1/paraRho + paraGamma;
64
65 % initialize the iterates for the fast gradient scheme
66 p = zeros(n,m);
```

```
67  w = p;
68
69  % initialize the current number of iteration
70  nIterations = 0;
71
72  % apply the fast gradient scheme for the specified
73  % maximal number of iterations
74
75  while ( nIterations < maxIterations )
76    nIterations = nIterations + 1;
77
78    % calculate the proximal point of f
79    proxF = zeros(n,m);
80    v = -imfilter(w,fsp,'conv','symmetric');
81    for i=1:n
82      for j=1:m
83        if ( v(i,j) < -paraLambda )
84          proxF(i,j) = (v(i,j)+paraLambda)/paraRho;
85        elseif ( v(i,j) > paraLambda )
86          proxF(i,j) = ( v(i,j) - paraLambda )/paraRho;
87        else
88          proxF(i,j) = 0;
89        end
90      end
91    end
92
93    % project the values onto the feasible set S
94    for i=1:n
95      for j=1:m
96        if ( proxF(i,j) < S(1) )
97          proxF(i,j) = S(1);
98        elseif ( proxF(i,j) > S(2) )
99          proxF(i,j) = S(2);
100       end
101     end
102   end
103
104   % calculate the proximal point of g
105   proxG = (2.*noisyB + w)./(2+paraMu);
106
107   % project the values onto the feasible set S
108   for i=1:n
109     for j=1:m
110       if ( proxG(i,j) < S(1) )
```

```
111           proxG(i,j) = S(1);
112         elseif ( proxG(i,j) > S(2) )
113           proxG(i,j) = S(2);
114         end
115       end
116     end
117
118     % calculate the gradient of the doubly smoothed objective
119     gradientFds=proxG-imfilter(proxF,fsp,'conv','symmetric')...
120                   + paraGamma.*w;
121
122     % update the iterates
123     p_new = w - ( 1/L )*gradientFds;
124     w_new = p_new + ( (sqrt(L) - sqrt(paraGamma))/ ...
125             ( sqrt(L) + sqrt(paraGamma)) )*(p_new-p);
126
127     p = p_new;
128     w = w_new;
129 end
130
131 % calculate the proximal point w.r.t. the resulting
132 % iterate p
133 proxFp = zeros(n,m);
134 v = -imfilter(p,fsp,'conv','symmetric');
135 for i=1:n
136   for j=1:m
137     if ( v(i,j) < -paraLambda )
138       proxFp(i,j) = (v(i,j)+paraLambda)/paraRho;
139     elseif ( v(i,j) > paraLambda )
140       proxFp(i,j) = ( v(i,j) - paraLambda )/paraRho;
141     else
142       proxFp(i,j) = 0;
143     end
144   end
145 end
146
147 % show the restoration of the image
148 figure('Name','restored')
149 imshow(proxFp.*( 1/S(2)*255 ))
```

A.1.2 Text test image

Matlab source code for solving the image restoration task described in Subsection 5.1.2 for the text test image. The m-file containing this source code can be found

on the compact disk attached to this thesis (*SourceCode/ImageRestoration/TextImageRestoration.m*).

```
% initialize parameters
noiseFac    = 0.001; % factor for additive noise
R           = 0.05;  % upper bound for the norm of dual
                     % solution
paraLambda = 2e-6;  % regularization parameter
epsi        = 0.1;   % accuracy for the double smoothing
                     % algorithm

% specify the number of iterations
maxIterations = 150;

% read the image
[pdata,~] = imread('text.png');

% show the original image
figure('Name','original image')
imshow(pdata)

% convert the data into double format
X = double(pdata);

% initialize the Gaussian lowpass filter of size 9 times 9
% and standard deviation 4
fsp = fspecial('gaussian',[9 9],4);

% blurr the scaled original image
B = imfilter(X,fsp,'conv','symmetric');

% initialize the noise matrix to be added to the blurred
% image B and add the noise
rng(57977,'v4');
noise = noiseFac*randn(size(X));

noisyB = B + noise;

% show the blurred and noisy image
figure('Name','blurred + noisy');
imshow(noisyB)

% get the dimensions of the image
[n,m] = size(noisyB);

```

```
43 % specify the feasible set
44 S = [0,1];
45
46 % specify the constants D_f and D_g
47 Df = 32768;
48 Dg = 32768;
49
50 % specify the smoothing parameters
51 paraRho   = epsi/(4*Df);
52 paraMu    = epsi/(4*Dg);
53 paraGamma = (2*epsi)/(4*R^2);
54
55 % specify the Lipschitz constant of the gradient
56 L = 1/paraMu + 1/paraRho + paraGamma;
57
58 % initialize the iterates for the fast gradient scheme
59 p = zeros(n,m);
60 w = p;
61
62 % initialize the current number of iteration
63 nIterations = 0;
64
65 % apply the fast gradient scheme for the specified
66 % maximal number of iterations
67 while ( nIterations < maxIterations )
68   nIterations = nIterations + 1;
69
70   % calculate the proximal point of f
71   proxF = zeros(n,m);
72   v = -imfilter(w,fsp,'conv','symmetric');
73   for i=1:n
74     for j=1:m
75       if ( v(i,j) < -paraLambda )
76         proxF(i,j) = (v(i,j)+paraLambda)/paraRho;
77       elseif ( v(i,j) > paraLambda )
78         proxF(i,j) = ( v(i,j) - paraLambda )/paraRho;
79       else
80         proxF(i,j) = 0;
81       end
82     end
83   end
84
85   % project the values onto the feasible set S
86   for i=1:n
```

```
87      for j=1:m
88        if ( proxF(i,j) < S(1) )
89          proxF(i,j) = S(1);
90        elseif ( proxF(i,j) > S(2) )
91          proxF(i,j) = S(2);
92        end
93      end
94    end
95
96    % calculate the proximal point of g
97    proxG = (2.*noisyB + w)./(2+paraMu);
98
99    % project the values onto the feasible set S
100   for i=1:n
101     for j=1:m
102       if ( proxG(i,j) < S(1) )
103         proxG(i,j) = S(1);
104       elseif ( proxG(i,j) > S(2) )
105         proxG(i,j) = S(2);
106       end
107     end
108   end
109
110   % calculate the gradient of the doubly smoothed objective
111   gradientFds=proxG-imfilter(proxF,fsp,'conv','symmetric')...
112                 + paraGamma.*w;
113
114   % update the iterates
115   p_new = w - ( 1/L )*gradientFds;
116   w_new = p_new + ( (sqrt(L) - sqrt(paraGamma))/ ...
117           ( sqrt(L) + sqrt(paraGamma)) )*(p_new-p);
118
119   p = p_new;
120   w = w_new;
121 end
122
123 % calculate the proximal point w.r.t. the resulting
124 % iterate p
125 proxFp = zeros(n,m);
126 v = -imfilter(p,fsp,'conv','symmetric');
127 for i=1:n
128   for j=1:m
129     if ( v(i,j) < -paraLambda )
130       proxFp(i,j) = (v(i,j)+paraLambda)/paraRho;
```

```
131       elseif ( v(i,j) > paraLambda )
132         proxFp(i,j) = ( v(i,j) - paraLambda )/paraRho;
133       else
134         proxFp(i,j) = 0;
135       end
136     end
137 end
138 
139 % show the restoration of the image
140 figure('Name','restored')
141 imshow(proxFp)
```

A.1.3 Cameraman test image

Matlab source code for solving the image restoration task described in Subsection 5.1.2 for the cameraman test image for the case of two functions in the objective. The m-file containing this source code can be found on the compact disk attached to this thesis (*SourceCode/ImageRestoration/CameramanImageRestoration2Functions.m*).

```
1  % initialize parameters
2  noiseFac    = 0.001; % factor for additive noise
3  R           = 0.05;  % upper bound for the norm of dual
4                       % solution
5  paraLambda = 2e-6;  % regularization parameter
6  epsi        = 0.05;  % accuracy for the double smoothing
7                       % algorithm
8  
9  % specify the number of iterations
10 maxIterations = 200;
11 
12 % read the image
13 [pdata,~]=imread('cameraman.tif');
14 
15 % show the original image
16 figure('Name','original');
17 imshow(pdata);
18 
19 % specify the lower and upper bound of the interval the
20 % pixels have to be scaled to
21 S = [0,0.1];
22 
23 % change the format of the pixel data and scale image
24 % to [0,1]
```

```
25 X=double(pdata);
26 X=X./255;
27
28 % initialize the Gaussian lowpass filter of size 9 times 9
29 % and standard deviation 4
30 fsp=fspecial('gaussian',[9 9],4);
31
32 % blurr the scaled image and initialize the additive noise
33 B=imfilter(X,fsp,'conv','symmetric');
34 rng(57977,'v4');
35 noise=noiseFac * randn(size(X));
36
37 % add noise to the blurred image
38 noisyB = B + noise;
39
40 % show the blurred and noisy image
41 figure('Name','blurred + noisy');
42 imshow(noisyB)
43
44 % scale the blurred and noisy image
45 noisyB = S(2).*noisyB;
46
47 % get the dimensions of the image
48 [n,m] = size(noisyB);
49
50 % specify the constants D_f and D_g, calculated in advance
51 Df = 327.68;
52 Dg = 327.68;
53
54 % specify the smoothing parameters
55 paraRho   = epsi/(4*Df);
56 paraMu    = epsi/(4*Dg);
57 paraGamma = (2*epsi)/(4*R^2);
58
59 % calculate the Lipschitz constant
60 L = 1/paraMu + 1/paraRho + paraGamma;
61
62 % initialize the iterates to zero
63 p = zeros(n,m);
64 w = p;
65
66 % specify the current number of iterations
67 nIterations = 0;
68
```

```
% apply the fast gradient scheme for the specified
% maximal number of iterations
while ( nIterations < maxIterations )
  nIterations = nIterations + 1;

  % calculate the proximal point of f and project it
  % to the feasible interval
  proxF = (-imfilter(w,fsp,'conv','symmetric') - ...
           paraLambda)./paraRho;
  for i=1:n
    for j=1:m
      if ( proxF(i,j) < S(1) )
        proxF(i,j) = S(1);
      elseif ( proxF(i,j) > S(2) )
        proxF(i,j) = S(2);
      end
    end
  end

  % calculate the proximal point of g and project
  % it to the feasible interval
  proxG = zeros(size(noisyB));
  for i=1:n
    for j=1:m
      if ( w(i,j) < paraMu*noisyB(i,j) -1 )
        proxG(i,j) = ( w(i,j) + 1 )/paraMu;
      elseif ( w(i,j) > paraMu*noisyB(i,j) + 1 )
        proxG(i,j) = ( w(i,j) - 1 )/paraMu;
      else
        proxG(i,j) = noisyB(i,j);
      end

      if ( proxG(i,j) < S(1) )
        proxG(i,j) = S(1);
      elseif ( proxG(i,j) > S(2) )
        proxG(i,j) = S(2);
      end
    end
  end

  % calculate the gradient of the doubly smoothed
  % function
  gradientFds=proxG-imfilter(proxF,fsp,'conv','symmetric')...
              + paraGamma.*w;
```

```
113
114   % update the iterates
115   p_new = w - ( 1/L )*gradientFds;
116   w_new = p_new + ( (sqrt(L) - sqrt(paraGamma))/...
117            ( sqrt(L) + sqrt(paraGamma)) )*(p_new-p);
118
119   p = p_new;
120   w = w_new;
121 end
122
123 % calculate the proximal point of f at the approximate
124 % optimal solution of the dual and project it to the
125 % feasible set
126 proxFp=(-imfilter(p,fsp,'conv','symmetric')-paraLambda)...
127          ./paraRho;
128 for i=1:n
129   for j=1:m
130     if ( proxFp(i,j) < S(1) )
131       proxFp(i,j) = S(1);
132     elseif ( proxFp(i,j) > S(2) )
133       proxFp(i,j) = S(2);
134     end
135   end
136 end
137
138 % show the restoration of the image
139 figure('Name','restored')
140 imshow(proxFp.*(1/S(2)*255))
```

Matlab source code for solving the image restoration task described in Subsection 5.1.2 for the cameraman test image for the case of three functions in the objective. The m-file containing this source code can be found on the compact disk attached to this thesis (*SourceCode/ImageRestoration/CameramanImageRestoration3Functions.m*).

```
1 % initialize parameters
2 noiseFac   = 0.001; % factor for additive noise
3 R          = 0.05;  % upper bound for the norm of dual
4                     % solution
5 paraLambda = 2e-6;  % regularization parameter
6 epsi       = 0.05;  % accuracy for the double smoothing
7                     % algorithm
8
9 % specify the number of iterations
```

```
10 maxIterations = 200;
11
12 % read the image
13 [pdata,~]=imread('cameraman.tif');
14
15 % show the original image
16 figure('Name','original');
17 imshow(pdata);
18
19 % specify the lower and upper bound of the interval the
20 % pixels have to be scaled to
21 S = [0,0.1];
22
23 % change the format of the pixel data and sclae image
24 % to [0,1]
25 X=double(pdata);
26 X=X./255;
27
28 % initialize the Gaussian lowpass filter of size 9 times 9
29 % and standard deviation 4
30 fsp=fspecial('gaussian',[9 9],4);
31
32 % blurr the scaled image and initialize the additive noise
33 B=imfilter(X,fsp,'conv','symmetric');
34 rng(57977,'v4');
35 noise=noiseFac * randn(size(X));
36
37 % add noise to the blurred image
38 noisyB = B + noise;
39
40 % show the blurred and noisy image
41 figure('Name','blurred + noisy');
42 imshow(noisyB)
43
44 % scale the blurred and noisy image
45 noisyB = S(2).*noisyB;
46
47 % get the dimensions of the image
48 [n,m] = size(noisyB);
49
50 % specify the constants D_f, D_g1 and D_g2
51 Df  = 327.68;
52 Dg1 = 327.68;
53 Dg2 = 327.68;
```

```
54
55 % specify the smoothing parameters
56 paraRho   = epsi/(5*Df);
57 paraMu1   = epsi/(5*Dg1);
58 paraMu2   = epsi/(5*Dg2);
59 paraGamma = (2*epsi)/(5*R^2);
60
61 % calculate the Lipschitz constant
62 L = 1/paraMu1 + 1/paraRho + paraGamma;
63
64 % initialize the iterates to zero
65 p = zeros(2*n,m);
66 w = p;
67
68 % specify the current number of iterations
69 nIterations = 0;
70
71 % apply the fast gradient scheme for the specified
72 % maximal number of iterations
73 while ( nIterations < maxIterations )
74   nIterations = nIterations + 1;
75
76   % calculate the proximal point of f and project the
77   % values to the feasible set
78   w1 = w(1:n,:);
79   w2 = w(n+1:2*n,:);
80   proxF = (-imfilter(w1+w2,fsp,'conv','symmetric') ...
81           - paraLambda)./paraRho;
82   for i=1:n
83     for j=1:m
84       if ( proxF(i,j) < S(1) )
85         proxF(i,j) = S(1);
86       elseif ( proxF(i,j) > S(2) )
87         proxF(i,j) = S(2);
88       end
89     end
90   end
91
92   % calculate the proximal point of g1 || Ax - b ||^2
93   % and project the values to the feasible set
94   proxG1 = (2.*noisyB + w1)./(2+paraMu1);
95   for i=1:n
96     for j=1:m
97       if ( proxG1(i,j) < S(1) )
```

```
98          proxG1(i,j) = S(1);
99        elseif ( proxG1(i,j) > S(2) )
100         proxG1(i,j) = S(2);
101       end
102     end
103   end
104
105   % calculate the proximal point of g2 || Ax - b ||_1
106   % and project the values to the feasible set
107   proxG2 = zeros(size(noisyB));
108   for i=1:n
109     for j=1:m
110       if ( w2(i,j) < paraMu2*noisyB(i,j) -1 )
111         proxG2(i,j) = ( w2(i,j) + 1 )/paraMu2;
112       elseif ( w2(i,j) > paraMu2*noisyB(i,j) + 1 )
113         proxG2(i,j) = ( w2(i,j) - 1 )/paraMu2;
114       else
115         proxG2(i,j) = noisyB(i,j);
116       end
117
118       if ( proxG2(i,j) < S(1) )
119         proxG2(i,j) = S(1);
120       elseif ( proxG2(i,j) > S(2) )
121         proxG2(i,j) = S(2);
122       end
123     end
124   end
125
126   % calculate the gradient of the double smoothed
127   % objective
128   KproxF = imfilter(proxF,fsp,'conv','symmetric');
129   gradientFs  = [proxG1;proxG2] - [KproxF;KproxF];
130   gradientFds = gradientFs + paraGamma.*[w1;w2];
131
132   % update the iterates
133   p_new = w - ( 1/L )*gradientFds;
134   w_new = p_new + ( (sqrt(L) - sqrt(paraGamma))/...
135           ( sqrt(L) + sqrt(paraGamma)) )*(p_new-p);
136
137   p = p_new;
138   w = w_new;
139 end
140
141 % calculate the proximal point of f at the resulting
```

```
142 % approximate dual solution
143 p1 = p(1:n,:);
144 p2 = p(n+1:2*n,:);
145
146 proxFp = (-imfilter(p1+p2,fsp,'conv','symmetric') ...
147           - paraLambda)./paraRho;
148 for i=1:n
149   for j=1:m
150     if ( proxFp(i,j) < S(1) )
151       proxFp(i,j) = S(1);
152     elseif ( proxFp(i,j) > S(2) )
153       proxFp(i,j) = S(2);
154     end
155   end
156 end
157
158 % show the restoration of the image
159 figure('Name','restored')
160 imshow(proxFp.*(1/S(2)*255))
```

A.2 Support vector regression source code

This section contains the source codes for the support vector regression tasks solved via the double smoothing approach introduced in Subsection 5.2. First we present the function `createTestData` that has as input the kernel parameter and the noise value and as output the training data set sampled from the sinc function. It returns an array containing the input points ($x-$values) and an array containing the corresponding values ($y-$values). Furthermore, the kernel matrix w. r. t. the samples is computed and returned. This function is called by all implementations of the regression task for the different loss functions.

The m-file containing this source code can be found on the compact disk attached to this thesis (*SourceCode/Regression/createTestData.m*).

```
1 function [xIn,yIn,K] = createTestData(sig,noise)
2
3   x = -5:0.01:5;
4   trueSinc = sinc(x);
5
6   xInput = x(1):0.2:x(end);
7   xInput = xInput';
8   yInput = zeros(length(xInput),1);
9
```

```
10    for i = 1:length(yInput)
11      yInput(i) = normrnd(sinc(xInput(i)),noise);
12    end
13    n = length(xInput);
14
15    % calculate the kernel matrix
16    K = zeros(n,n);
17    for i=1:n
18      for j=1:n
19          K(i,j) = exp(-( norm(xInput(i) - xInput(j))^2)...
20                    /(2*sig^2) );
21      end
22    end
23    % plot sample points together with sinc function
24    figure('Position',[50 50 1000 850]);
25    axis([min(x)-0.1 max(x)+0.1 -0.6 1.2]);
26    hold on
27    plot(x,trueSinc,'LineWidth',2);
28    plot(xInput,yInput,'o');
29    legend3 = legend('f(x)=sinc(x)','training data');
30    set(legend3,'FontSize',12);
```

A.2.1 The ε−insensitive loss function

Matlab source code for solving the support vector regression task w.r.t. the ε−insensitive loss function (cf. Subsection 5.2.2). The m-file containing this source code can be found on the compact disk attached to this thesis (*SourceCode/Regression/RegressionEps.m*).

```
1  % specifiy the number of iterations
2  nIterations = 20000;
3
4  % SVR parameters
5  svSigma = 0.5;  % kernel parameter
6  svC     = 1;    % regularization parameter
7  svEpsi  = 0.05; % epsilon for the loss function
8  svXi    = 0.15; % upper bound on allowed deviation
9
10 % specify the noise value and create training data
11 noise = 0.05;
12 [xIn,yIn,K] = createTestData(svSigma,noise);
13
14 m = length(xIn);
15
```

```
% DS parameters
dsEpsi = 0.05; % accuracy for the DS algorithm
dsR    = 3.4;  % upper bound on norm of dual solution
dsGamma= ( 2*dsEpsi )/( (m+2)*dsR^2 );% smoothing parameter

dsMu = zeros(m,1);
Dg = zeros(m,1);

% determine the maximum of the smoothing parameters
% mu_i, i=1,...,m
maxVal = zeros(m,1);
for i=1:m
  if ( yIn(i) >= 0 )
    maxVal(i) = 0.5*(yIn(i) + svXi)^2;
  else
    maxVal(i) = 0.5*(yIn(i) - svXi)^2;
  end
end

sumMaxVal = sum(maxVal);
for i=1:m
  Dg(i) = sumMaxVal;
  dsMu(i) = dsEpsi/( (m+2)*Dg(i) );
end

maxDsMu = 1/dsMu(1);
for i=2:m
  if ( 1/dsMu(i) > maxDsMu )
    maxDsMu = 1/dsMu(i);
  end
end

% calculate the Lipschitz constant
dsL = maxDsMu + m*norm(K) + dsGamma;

% initialize the iterates
w = zeros(m,m);
p = zeros(m,m);

% apply the double smoothing algorithm for the
% specified number of iterations
for nIter = 1:nIterations

  % calculate the gradient of f
```

```
60  sumW = zeros(m,1);
61  for i=1:m
62    sumW = sumW + w(:,i);
63  end
64  gradF = -sumW;
65
66  arrayOfGrad = zeros(m,m);
67  for i=1:m
68    arrayOfGrad(:,i) = gradF;
69  end
70
71  % calculate the proximal points for all g_i,
72  % i=1,...,m
73  proxG = zeros(m,m);
74  % loop over columns
75  for i=1:m
76    % loop over rows
77    for j=1:m
78      if ( i==j )
79        if ( w(j,i) < dsMu(i)*(yIn(i) - svEpsi) - svC )
80          proxG(j,i) = ( w(j,i) + svC )/dsMu(i);
81        elseif ( w(j,i) >= dsMu(i)*(yIn(i)-svEpsi)-svC ...
82              && w(j,i) <= dsMu(i)*(yIn(i) - svEpsi) )
83          proxG(j,i) = yIn(i) - svEpsi;
84        elseif ( w(j,i) > dsMu(i)*( yIn(i) - svEpsi ) ...
85              && w(j,i) < dsMu(i)*( yIn(i) + svEpsi ) )
86          proxG(j,i) = w(j,i)/dsMu(i);
87        elseif ( w(j,i) >= dsMu(i)*( yIn(i)+svEpsi ) ...
88              && w(j,i) <= dsMu(i)*(yIn(i)+svEpsi)+svC)
89          proxG(j,i) = yIn(i) + svEpsi;
90        else
91          proxG(j,i) = ( w(j,i) - svC )/dsMu(i);
92        end
93      else
94        proxG(j,i) = w(j,i)/dsMu(i);
95      end
96    end
97  end
98
99  % projection
100 for i=1:m
101   for j=1:m
102     if ( proxG(j,i) <= yIn(j) - svXi )
103       proxG(j,i) = yIn(j) - svXi;
```

```
104         elseif ( proxG(j,i) >= yIn(j) + svXi )
105           proxG(j,i) = yIn(j) + svXi;
106         end
107       end
108     end
109
110     % calculate the gradient of the doubly smoothed
111     % objective
112     Z = zeros(m,m);
113     for i=1:m
114       Z(:,i) = -K*arrayOfGrad(:,i);
115     end
116     gradientFs = proxG + Z;
117     gradientFds = gradientFs + dsGamma.*w;
118
119     % update the iterates
120     p_new = w - ( 1/dsL ).*gradientFds;
121     w_new = p_new + ( ( sqrt(dsL) - sqrt(dsGamma) )/ ...
122             ( sqrt(dsL) + sqrt(dsGamma) ) ).*(p_new - p);
123
124     p = p_new;
125     w = w_new;
126 end
127
128 % calculate the gradient of f at resulting approximate
129 % dual solution
130 sumGradFp = zeros(m,1);
131 for i=1:m
132     sumGradFp = sumGradFp + p(:,i);
133 end
134
135 % the resulting expansion coefficients
136 c = -sumGradFp;
137
138 % calculate the true sinc function and the
139 % resulting regression function and plot the
140 % results
141 x = -5:0.05:5;
142 n = length(x);
143 yPredict = zeros(n,1);
144 ySinc = sinc(x);
145
146 for i=1:n
147     for j=1:m
```

```
148             yPredict(i) = yPredict(i) + c(j)*exp( ...
149                 -( norm(x(i)-xIn(j)) )^2/(2*svSigma^2) );
150         end
151     end
152
153     figure('Name','Result')
154     hold on
155     plot(x,ySinc,'LineWidth',1);
156     plot(x,yPredict,'r','LineWidth',2);
157     plot(xIn,yIn,'.','MarkerSize',20);
158     plot(x,(yPredict+svXi),'r--','LineWidth',1);
159     plot(x,(yPredict-svXi),'r--','LineWidth',1);
160     plot(x,(yPredict+svEpsi),'r-.','LineWidth',1);
161     plot(x,(yPredict-svEpsi),'r-.','LineWidth',1);
162     legend('true sinc','predicted','input');
163     hold off
```

A.2.2 The quadratic ε−insensitive loss function

Matlab source code for solving the support vector regression task w.r.t. the quadratic ε−insensitive loss function (cf. Subsection 5.2.2). The m-file containing this source code can be found on the compact disk attached to this thesis (*SourceCode/Regression/RegressionEpsQuadrat.m*).

```
1   % specifiy the number of iterations
2   nIterations = 20000;
3
4   % SVR parameters
5   svSigma = 0.5;  % kernel parameter
6   svC     = 1;    % regularization parameter
7   svEpsi  = 0.05; % epsilon for the loss function
8   svXi    = 0.15; % upper bound on allowed deviation
9
10  % specify the noise value and create training data
11  noise = 0.05;
12  [xIn,yIn,K] = createTestData(svSigma,noise);
13
14  m = length(xIn);
15
16  % DS parameters
17  dsEpsi  = 0.05; % accuracy for the DS algorithm
18  dsR     = 0.5;  % upper bound on norm of dual solution
19  dsGamma = (2*dsEpsi)/((m+2)*dsR^2); % smoothing parameter
20
```

```
21 dsMu = zeros(m,1);
22 Dg = zeros(m,1);
23
24 % determine the maximum of the smoothing parameters
25 % mu_i, i=1,...,m
26 maxVal = zeros(m,1);
27 for i=1:m
28   if ( yIn(i) >= 0 )
29     maxVal(i) = 0.5*(yIn(i) + svXi)^2;
30   else
31     maxVal(i) = 0.5*(yIn(i) - svXi)^2;
32   end
33 end
34
35 sumMaxVal = sum(maxVal);
36 for i=1:m
37   Dg(i) = sumMaxVal;
38   dsMu(i) = dsEpsi/( (m+2)*Dg(i) );
39 end
40
41 maxDsMu = 1/dsMu(1);
42 for i=2:m
43   if ( 1/dsMu(i) > maxDsMu )
44     maxDsMu = 1/dsMu(i);
45   end
46 end
47
48 % calculate the Lipschitz constant
49 dsL = maxDsMu + m*norm(K) + dsGamma;
50
51 % initialize the iterates
52 w = zeros(m,m);
53 p = zeros(m,m);
54
55 % apply the double smoothing algorithm for the specified
56 % number of iterations
57 for nIter = 1:nIterations
58
59   % calculate the gradient of f
60   sumW = zeros(m,1);
61   for i=1:m
62     sumW = sumW + w(:,i);
63   end
64   gradF = -sumW;
```

```
65
66  arrayOfGrad = zeros(m,m);
67  for i=1:m
68    arrayOfGrad(:,i) = gradF;
69  end
70
71  % calculate the proximal points for all g_i,
72  % i=1,...,m
73  proxG = zeros(m,m);
74  % loop over columns
75  for i=1:m
76    % loop over rows
77    for j=1:m
78      if ( i==j )
79        if ( w(j,i) < dsMu(i)*( yIn(i) -svEpsi ) )
80          proxG(j,i) = ( 2*svC*( yIn(i) - svEpsi ) ...
81                         + w(j,i) )/( 2*svC + dsMu(i) );
82        elseif ( w(j,i) >= dsMu(i)*( yIn(i) - svEpsi ) ...
83              && w(j,i) <= dsMu(i)*( yIn(i) + svEpsi ) )
84          proxG(j,i) = w(j,i)/dsMu(i);
85        elseif ( w(j,i) > dsMu(i)*( yIn(i) + svEpsi ) )
86          proxG(j,i) = ( 2*svC*( yIn(i) + svEpsi ) ...
87                           + w(j,i) )/( 2*svC + dsMu(i) );
88        end
89      else
90        proxG(j,i) = w(j,i)/dsMu(i);
91      end
92    end
93  end
94
95  % projection
96  for i=1:m
97    for j=1:m
98      if ( proxG(j,i) <= yIn(j) - svXi )
99        proxG(j,i) = yIn(j) - svXi;
100     elseif ( proxG(j,i) >= yIn(j) + svXi )
101       proxG(j,i) = yIn(j) + svXi;
102     end
103   end
104 end
105
106 % calculate the gradient of the doubly smoothed
107 % objective
108 Z = zeros(m,m);
```

```
109     for i=1:m
110       Z(:,i) = -K*arrayOfGrad(:,i);
111     end
112     gradientFs = proxG + Z;
113     gradientFds = gradientFs + dsGamma.*w;
114
115     % update the iterates
116     p_new = w - ( 1/dsL ).*gradientFds;
117     w_new = p_new + ( ( sqrt(dsL) - sqrt(dsGamma) )/ ...
118             ( sqrt(dsL) + sqrt(dsGamma) ) ).*(p_new - p);
119
120     p = p_new;
121     w = w_new;
122 end
123
124 % calculate the gradient of f at resulting approximate
125 % dual solution
126 sumGradFp = zeros(m,1);
127 for i=1:m
128     sumGradFp = sumGradFp + p(:,i);
129 end
130
131 % the resulting expansion coefficients
132 c = -sumGradFp;
133
134 % calculate the true sinc function and the
135 % resulting regression function and plot the
136 % results
137 x = -5:0.05:5;
138 n = length(x);
139 yPredict = zeros(n,1);
140 ySinc = sinc(x);
141
142 for i=1:n
143     for j=1:m
144         yPredict(i) = yPredict(i) + c(j)*exp( ...
145               -( norm(x(i)-xIn(j)) )^2/(2*svSigma^2) );
146     end
147 end
148
149 figure('Name','Result')
150 hold on
151 plot(x,ySinc,'LineWidth',1);
152 plot(x,yPredict,'r','LineWidth',2);
```

```
153 plot(xIn,yIn,'.','MarkerSize',20);
154 plot(x,(yPredict+svXi),'r--','LineWidth',1);
155 plot(x,(yPredict-svXi),'r--','LineWidth',1);
156 plot(x,(yPredict+svEpsi),'r-.','LineWidth',1);
157 plot(x,(yPredict-svEpsi),'r-.','LineWidth',1);
158 legend('true sinc','predicted','input');
159 hold off
```

A.2.3 The Huber loss function

Matlab source code for solving the support vector regression task w.r.t. the Huber loss function (cf. Subsection 5.2.2). The m-file containing this source code can be found on the compact disk attached to this thesis (*SourceCode/Regression/RegressionHuber.m*).

```
1  % specifiy the number of iterations
2  nIterations = 20000;
3
4  % SVR parameters
5  svSigma = 0.5;  % kernel parameter
6  svC     = 10;    % regularization parameter
7  svEpsi  = 0.05; % epsilon for the loss function
8  svXi    = 0.15; % upper bound on allowed deviation
9
10 % specify the noise value and create training data
11 noise = 0.05;
12 [xIn,yIn,K] = createTestData(svSigma,noise);
13
14 m = length(xIn);
15
16 % DS parameters
17 dsEpsi  = 0.05; % accuracy for the DS algorithm
18 dsR     = 3;  % upper bound on norm of dual solution
19 dsGamma = ( 2*dsEpsi )/( (m+2)*dsR^2 ); % smoothing parameter
20
21 dsMu = zeros(m,1);
22 Dg = zeros(m,1);
23
24 % determine the maximum of the smoothing parameters
25 % mu_i, i=1,...,m
26 maxVal = zeros(m,1);
27 for i=1:m
28   if ( yIn(i) >= 0 )
29     maxVal(i) = 0.5*(yIn(i) + svXi)^2;
```

```
30      else
31        maxVal(i) = 0.5*(yIn(i) - svXi)^2;
32      end
33    end
34
35    sumMaxVal = sum(maxVal);
36    for i=1:m
37      Dg(i) = sumMaxVal;
38      dsMu(i) = dsEpsi/( (m+2)*Dg(i) );
39    end
40
41    maxDsMu = 1/dsMu(1);
42    for i=2:m
43      if ( 1/dsMu(i) > maxDsMu )
44        maxDsMu = 1/dsMu(i);
45      end
46    end
47
48    % calculate the Lipschitz constant
49    dsL = maxDsMu + m*norm(K) + dsGamma;
50
51    % initialize the iterates
52    w = zeros(m,m);
53    p = zeros(m,m);
54
55    % apply the double smoothing algorithm for the specified
56    % number of iterations
57    for nIter = 1:nIterations
58
59      % calculate the gradient of f
60      sumW = zeros(m,1);
61      for i=1:m
62        sumW = sumW + w(:,i);
63      end
64      gradF = -sumW;
65
66      arrayOfGrad = zeros(m,m);
67      for i=1:m
68        arrayOfGrad(:,i) = gradF;
69      end
70
71      % calculate the proximal points for all g_i,
72      % i=1,...,m
73      proxG = zeros(m,m);
```

```
74  % loop over columns
75  for i=1:m
76    % loop over rows
77    for j=1:m
78      if ( i==j )
79        if ( w(j,i) < dsMu(i)*(yIn(i)-svEpsi)-svC*svEpsi)
80          proxG(j,i) = ( svC*svEpsi + w(j,i) )/( dsMu(i) );
81        elseif (w(j,i)>=dsMu(i)*(yIn(i)-svEpsi)-svC*svEpsi...
82                && w(j,i)<=dsMu(i)*(yIn(i)+svEpsi)+svC*svEpsi)
83          proxG(j,i) = (svC*yIn(i)+w(j,i) )/(svC+dsMu(i));
84        else
85          proxG(j,i) = ( -svC*svEpsi + w(j,i) )/( dsMu(i) );
86        end
87      else
88        proxG(j,i) = w(j,i)/dsMu(i);
89      end
90    end
91  end
92
93  % projection
94  for i=1:m
95    for j=1:m
96      if ( proxG(j,i) <= yIn(j) - svXi )
97        proxG(j,i) = yIn(j) - svXi;
98      elseif ( proxG(j,i) >= yIn(j) + svXi )
99        proxG(j,i) = yIn(j) + svXi;
100     end
101   end
102 end
103
104 % calculate the gradient of the doubly smoothed function
105 Z = zeros(m,m);
106 for i=1:m
107   Z(:,i) = -K*arrayOfGrad(:,i);
108 end
109 gradientFs = proxG + Z;
110 gradientFds = gradientFs + dsGamma.*w;
111
112 % update the iterates
113 p_new = w - ( 1/dsL ).*gradientFds;
114 w_new = p_new + ( ( sqrt(dsL) - sqrt(dsGamma) )/ ...
115         ( sqrt(dsL) + sqrt(dsGamma) ) ).*(p_new - p);
116
117 p = p_new;
```

```
118   w = w_new;
119 end
120
121 % calculate the gradient of f at resulting approximate
122 % dual solution
123 sumGradFp = zeros(m,1);
124 for i=1:m
125     sumGradFp = sumGradFp + p(:,i);
126 end
127
128 % the resulting expansion coefficients
129 c = -sumGradFp;
130
131 % calculate the true sinc function and the
132 % resulting regression function and plot the
133 % results
134 x = -5:0.05:5;
135 n = length(x);
136 yPredict = zeros(n,1);
137 ySinc = sinc(x);
138
139 for i=1:n
140     for j=1:m
141         yPredict(i) = yPredict(i) + c(j)*exp( ...
142               -( norm(x(i)-xIn(j)) )^2/(2*svSigma^2) );
143     end
144 end
145
146 figure('Name','Result')
147 hold on
148 plot(x,ySinc,'LineWidth',1);
149 plot(x,yPredict,'r','LineWidth',2);
150 plot(xIn,yIn,'.','MarkerSize',20);
151 plot(x,(yPredict+svXi),'r--','LineWidth',1);
152 plot(x,(yPredict-svXi),'r--','LineWidth',1);
153 legend('true sinc','predicted','input');
154 hold off
```

A.2.4 The extended loss function

Matlab source code for solving the support vector regression task w. r. t. the extended loss function (cf. Subsection 5.2.2). The m-file containing this source code can be found on the compact disk attached to this thesis (*SourceCode/Regression/RegressionExtended.m*).

```
% specifiy the number of iterations
nIterations = 20000;

% SVR parameters
svSigma = 0.5;  % kernel parameter
svC     = 1;    % regularization parameter
svEpsi  = 0.1; % epsilon for the loss function
svXi    = svEpsi; % upper bound on allowed deviation

% specify the noise value and create training data
noise = 0.05;
[xIn,yIn,K] = createTestData(svSigma,noise);

m = length(xIn);

% DS parameters
dsEpsi  = 0.05; % accuracy for the DS algorithm
dsR     = 0.5;  % upper bound on norm of dual solution
dsGamma = (2*dsEpsi)/((m+2)*dsR^2); % smoothing parameter

dsMu = zeros(m,1);
Dg = zeros(m,1);

% determine the maximum of the smoothing parameters
% mu_i, i=1,...,m
maxVal = zeros(m,1);
for i=1:m
  if ( yIn(i) >= 0 )
    maxVal(i) = 0.5*(yIn(i) + svXi)^2;
  else
    maxVal(i) = 0.5*(yIn(i) - svXi)^2;
  end
end

sumMaxVal = sum(maxVal);
for i=1:m
  Dg(i) = sumMaxVal;
  dsMu(i) = dsEpsi/( (m+2)*Dg(i) );
end

maxDsMu = 1/dsMu(1);
for i=2:m
  if ( 1/dsMu(i) > maxDsMu )
    maxDsMu = 1/dsMu(i);
```

```
45    end
46  end
47
48  % calculate the Lipschitz constant
49  dsL = maxDsMu + m*norm(K) + dsGamma;
50
51  % initialize the iterates
52  w = zeros(m,m);
53  p = zeros(m,m);
54
55  % apply the double smoothing algorithm for the specified
56  % number of iterations
57  for nIter = 1:nIterations
58
59    % calculate the gradient of f
60    sumW = zeros(m,1);
61    for i=1:m
62      sumW = sumW + w(:,i);
63    end
64    gradF = -sumW;
65
66    arrayOfGrad = zeros(m,m);
67    for i=1:m
68      arrayOfGrad(:,i) = gradF;
69    end
70
71    % calculate the proximal points for all g_i,
72    % i=1,...,m
73    proxG = zeros(m,m);
74    % loop over columns
75    for i=1:m
76      % loop over rows
77      for j=1:m
78        proxG(j,i) = w(j,i)/dsMu(i);
79      end
80    end
81
82    % projection
83    for i=1:m
84      for j=1:m
85        if ( proxG(j,i) <= yIn(j) - svXi )
86          proxG(j,i) = yIn(j) - svXi;
87        elseif ( proxG(j,i) >= yIn(j) + svXi )
88          proxG(j,i) = yIn(j) + svXi;
```

```
89          end
90        end
91      end
92
93      % calculate the gradient of the doubly smoothed
94      % dual objective
95      Z = zeros(m,m);
96      for i=1:m
97        Z(:,i) = -K*arrayOfGrad(:,i);
98      end
99      gradientFs = proxG + Z;
100     gradientFds = gradientFs + dsGamma.*w;
101
102     % update the iterates
103     p_new = w - ( 1/dsL ).*gradientFds;
104     w_new = p_new + ( ( sqrt(dsL) - sqrt(dsGamma) )/ ...
105             ( sqrt(dsL) + sqrt(dsGamma) ) ).*(p_new - p);
106
107     p = p_new;
108     w = w_new;
109 end
110
111 % calculate the gradient of f at resulting approximate
112 % dual solution
113 sumGradFp = zeros(m,1);
114 for i=1:m
115     sumGradFp = sumGradFp + p(:,i);
116 end
117
118 % the resulting expansion coefficients
119 c = -sumGradFp;
120
121 % calculate the true sinc function and the
122 % resulting regression function and plot the
123 % results
124 x = -5:0.05:5;
125 n = length(x);
126 yPredict = zeros(n,1);
127 ySinc = sinc(x);
128
129 for i=1:n
130     for j=1:m
131         yPredict(i) = yPredict(i) + c(j)*exp( ...
132             -( norm(x(i)-xIn(j)) )^2/(2*svSigma^2) );
```

```
133         end
134 end
135
136 figure('Name','Result')
137 hold on
138 plot(x,ySinc,'LineWidth',1);
139 plot(x,yPredict,'r','LineWidth',2);
140 plot(xIn,yIn,'.','MarkerSize',20);
141 plot(x,(yPredict+svXi),'r--','LineWidth',1);
142 plot(x,(yPredict-svXi),'r--','LineWidth',1);
143 legend('true sinc','predicted','input');
144 hold off
```

Theses

1. We consider the optimization problem

$$\left(\tilde{\mathrm{P}}_{\mathrm{SV}}\right) \qquad \inf_{c\in\mathbb{R}^n}\left\{C\sum_{i=1}^{n} v((Kc)_i, y_i) + \tilde{g}\left(\sqrt{c^{\mathrm{T}}Kc}\right)\right\},$$

where $n \in \mathbb{N}$ is the number of input data available, $v : \mathbb{R}\times\mathbb{R} \to \overline{\mathbb{R}}$ is proper and convex in its first variable, $K \in \mathbb{R}^{n\times n}$ is a real, symmetric and positive semidefinite kernel matrix and $\tilde{g} : \mathbb{R} \to \overline{\mathbb{R}}$ is defined by

$$\tilde{g}(t) = \begin{cases} g(t), & \text{if } t \geq 0, \\ +\infty, & \text{else,} \end{cases}$$

for the function $g : [0,\infty) \to \mathbb{R}$ assumed to be strictly monotonically increasing. This optimization problem arises from modeling the support vector machine problem for classification and regression, respectively, based on the Tikhonov regularization problem of finding the function f that is a minimizer of

$$\inf_{f\in\mathcal{H}_k}\left\{C\sum_{i=1}^{m} v(f(x_i), y_i) + g(\|f\|_{\mathcal{H}_k})\right\}$$

and is an element of the reproducing kernel Hilbert space $\mathcal{H}_k$ induced by a kernel function k that fulfills the finitely positive semidefiniteness property. The term $\|\cdot\|_{\mathcal{H}_k}$ denotes the norm in $\mathcal{H}_k$.

Due to the generalized representer theorem $\left(\tilde{\mathrm{P}}_{\mathrm{SV}}\right)$ is a generalized version of optimization problems for solving the support vector machines problem considered in most literature. This formulation allows the use of regularization terms other than the squared norm of the function $f \in \mathcal{H}_k$.

2. To the primal problem $\left(\tilde{\mathrm{P}}_{\mathrm{SV}}\right)$ we assign the Fenchel dual problem

$$\left(\tilde{\mathrm{D}}_{\mathrm{SV}}\right) \qquad \sup_{\substack{P\in\mathbb{R}^n \\ P=(P_1,\ldots,P_n)^{\mathrm{T}}}} \left\{ -C\sum_{i=1}^{n} \left(v(\cdot,y_i)\right)^* \left(-\frac{P_i}{C}\right) - \tilde{g}^* \left(\sqrt{P^{\mathrm{T}}KP}\right) \right\}$$

where $(v(\cdot,y_i))^*$ and $\tilde{g}^*$ denote the conjugate functions of $v(\cdot,y_i)$ and $\tilde{g}$, respectively, for all $i=1,\ldots,n$.

We show that weak duality always holds between the primal-dual pair $\left(\tilde{\mathrm{P}}_{\mathrm{SV}}\right) - \left(\tilde{\mathrm{D}}_{\mathrm{SV}}\right)$.

3. We employ the interior-point type qualification condition

$$(QC) \qquad \mathrm{Im}K \cap \prod_{i=1}^{n} \mathrm{ri}(\mathrm{dom}\, v(\cdot,y_i)) \neq \emptyset$$

and show that, whenever (QC) is fulfilled then strong duality holds between the primal-dual pair $\left(\tilde{\mathrm{P}}_{\mathrm{SV}}\right) - \left(\tilde{\mathrm{D}}_{\mathrm{SV}}\right)$ and derive necessary and sufficient optimality conditions.

4. For the particular case of K positive definite and the commonly used regularization term $\frac{1}{2}\|f\|^2_{\mathcal{H}_k}$ we derive several special primal and dual optimization problems when employing different loss functions for both the regression and the classification task. The resulting dual problems turn out to have differentiable objective functions and simple box constraints or nonnegativity constraints and thus are more easy to solve numerically compared to the corresponding primal problems.
We show how equivalent reformulations of the dual problems reveal the standard dual problems in the literature applying Lagrange duality instead of Fenchel-type duality.
We solve both the regression and the classification task numerically, the latter based on high dimensional real world data consisting of images and obtain an excellent performance of this approach.

5. We consider the general optimization problem

$$(\mathrm{P}_{\mathrm{gen}}) \qquad \inf_{x\in\mathbb{R}^n} \left\{ f(x) + \sum_{i=1}^{m} g_i(K_i x) \right\}$$

where $f:\mathbb{R}^n \to \overline{\mathbb{R}}$ is assumed to be proper, convex and lower semicontinuous. The operators $K_i:\mathbb{R}^n \to \mathbb{R}^{k_i}$ are assumed to be linear operators and

the functions $g_i : \mathbb{R}^{k_i} \to \overline{\mathbb{R}}$ are proper, convex and lower semicontinuous for all $i = 1, \ldots, m$, $m \in \mathbb{N}$. We assign the Fenchel dual problem

$$(\mathrm{D_{gen}}) \qquad \sup_{(p_1,\ldots,p_m)\in\mathbb{R}^{k_1}\times\mathbb{R}^{k_m}} \left\{ -f^* \left(-\sum_{i=1}^{m} K_i^* p_i \right) - \sum_{i=1}^{m} g_i^*(p_i) \right\}$$

to $(\mathrm{P_{gen}})$, where $f^* : \mathbb{R}^n \to \overline{\mathbb{R}}$ and $g_i^* : \mathbb{R}^{k_i} \to \overline{\mathbb{R}}$ are the conjugate functions of f and g_i, $i = 1, \ldots, m$, respectively. The operators $K_i^* : \mathbb{R}^{k_i} \to \mathbb{R}^n$ are the adjoint operators of K_i, $i = 1, \ldots, m$.
For the primal-dual pair $(\mathrm{P_{gen}}) - (\mathrm{D_{gen}})$ weak duality always holds. In order to prove that the optimal objective values $v\,(\mathrm{P_{gen}})$ and $v\,(\mathrm{D_{gen}})$ of the primal optimization problem $(\mathrm{P_{gen}})$ and the dual optimization problem $(\mathrm{D_{gen}})$ coincide and that the primal problem has an optimal solution we impose the interior-point regularity condition

$$(QC^*) \qquad 0 \in \mathrm{ri}\,(K^*(\mathrm{dom}\, g^*) + \mathrm{dom}\, f^*)\,.$$

Here, $K^* : \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m} \to \mathbb{R}^n$ is the adjoint operator of the operator $K : \mathbb{R}^n \to \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m}$ defined by $Kx = (K_1 x, \ldots, K_m x)$ and $g^* : \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m} \to \overline{\mathbb{R}}$ the conjugate function of the function $g : \mathbb{R}^{k_1} \times \ldots \times \mathbb{R}^{k_m} \to \overline{\mathbb{R}}$ defined by $g(y_1, \ldots, y_m) = \sum_{i=1}^m g_i(y_i)$. We show that in our setting this condition is always fulfilled and by proving strong duality between $(\mathrm{D_{gen}})$ and the Fenchel dual problem of $(\mathrm{D_{gen}})$ we get that there exists an optimal solution to $(\mathrm{P_{gen}})$ and $v\,(\mathrm{P_{gen}}) = v\,(\mathrm{D_{gen}})$.

6. In general, $(\mathrm{P_{gen}})$ is a nonsmooth optimization problem. In order to solve $(\mathrm{P_{gen}})$ approximately we develop a regularization technique that twice performs a smoothing w. r. t. the objective function of $(\mathrm{D_{gen}})$ as well as w. r. t. the objective function of the resulting single smoothed problem

$$(\mathrm{D}_{\rho,\mu}) \qquad \inf_{(p_1,\ldots,p_m)\in\mathbb{R}^{k_1}\times\mathbb{R}^{k_m}} \left\{ f_\rho^* \left(-\sum_{i=1}^{m} K_i^* p_i \right) + \sum_{i=1}^{m} g_{i,\mu_i}^*(p_i) \right\},$$

respectively, in order to efficiently solve $(\mathrm{D_{gen}})$ via a fast gradient algorithm. The objective function of $(\mathrm{D}_{\rho,\mu})$ is continuously differentiable with Lipschitz continuous gradient.

7. The doubly smoothed optimization problem that is actually solved is

$$(\mathrm{D}_{\rho,\mu,\gamma}) \qquad \inf_{(p_1,\ldots,p_m)\in\mathbb{R}^{k_1}\times\mathbb{R}^{k_m}} \left\{ f_\rho^* \left(-\sum_{i=1}^{m} K_i^* p_i \right) + \sum_{i=1}^{m} g_{i,\mu_i}^*(p_i) + \frac{\gamma}{2} \|(p_1, \ldots, p_m)\|_k^2 \right\}$$

and has a strongly convex and continuously differentiable objective function with Lipschitz continuous gradient whose Lipschitz constant is given by

$$L(\rho,\mu,\gamma) = \max_{i=1,\ldots,m}\left\{\frac{1}{\mu_i}\right\} + \frac{\sqrt{m}}{\rho}\left[\left(\sum_{i=1}^{m}\|K_i\|^2\right)\max_{i=1,\ldots,m}\left\{\|K_i\|^2\right\}\right]^{\frac{1}{2}} + \gamma$$

Thus, $(\mathrm{D}_{\rho,\mu,\gamma})$ can be solved efficiently by the fast gradient method.

8. For the sequence of dual variables $(p^k)_{k\geq 0} = (p_1^k,\ldots,p_m^k)_{k\geq 0} \subseteq \mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_m}$ generated by the fast gradient algorithm when solving $(\mathrm{D}_{\rho,\mu,\gamma})$ we show that for a given accuracy $\varepsilon > 0$ we can guarantee that

$$F(p^k) - F^* \leq \varepsilon$$

after a certain amount of iterations, where F is the objective function of $(\mathrm{D}_{\mathrm{gen}})$, $F^* = F(p^*)$ is the optimal objective value of $(\mathrm{D}_{\mathrm{gen}})$ and $p^* \in \mathbb{R}^{k_1}\times\ldots\times\mathbb{R}^{k_m}$ the optimal solution of $(\mathrm{D}_{\mathrm{gen}})$ which assumed to exist. Further it holds

$$\left\|\nabla F_{\rho,\mu}(p^k)\right\|_{\bar{k}} \leq \frac{\varepsilon}{R}$$

after a certain amount of iterations, where $F_{\rho,\mu}$ is the objective function of $(\mathrm{D}_{\rho,\mu})$ and $R > 0$ a constant such that $\|p^*\|_{\bar{k}} \leq R$.
We show that the above estimates hold after $k = O\left(\frac{1}{\varepsilon}\ln\left(\frac{1}{\varepsilon}\right)\right)$ iterations.

9. We show that one can construct an approximate optimal and feasible solution to $(\mathrm{P}_{\mathrm{gen}})$ with the help of the approximate solution to $(\mathrm{D}_{\mathrm{gen}})$. Furthermore, for a sequence $(\varepsilon_t)_{t\geq 0} \subseteq \mathbb{R}_+$ s. t. $\lim_{t\to\infty}\varepsilon_t = 0$ we show that the approximate optimal and feasible solution to $(\mathrm{P}_{\mathrm{gen}})$ converges to the optimal solution of $(\mathrm{P}_{\mathrm{gen}})$.

10. We apply the double smoothing technique to the problem of solving the support vector regression task in the case when f in the objective of $(\mathrm{P}_{\mathrm{gen}})$ is continuously differentiable and strongly convex. Then, the smoothing of f^* can be omitted and there is no need to require the domain of f to be a bounded set. In that case the rates of convergence are the same as in the general case of $(\mathrm{P}_{\mathrm{gen}})$ not assuming f to be differentiable and strongly convex.

11. We apply the double smoothing approach for solving image restoration tasks numerically and obtain that our approach performs better than FISTA on this task, at least w. r. t. the test images considered in this thesis.

We show that the double smoothing approach is suitable for numerically solving the support vector regression problem based on some toy data set. The proximal points needed to perform the regression task for different loss functions can easily be computed.

Bibliography

[1] M.V. Afonso, J. Bioucas-Dias, and M. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE Transactions on Image Processing*, 19(9):2345–2356, 2010.

[2] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 686:337–404, 1950.

[3] H. Attouch and M. Théra. A general duality principle for the sum of two operators. *Journal of Convex Analysis*, 3(1):1–24, 1996.

[4] H.H. Bauschke, R.I. Boţ, W.L. Hare, and W.M. Moursi. Attouch-Théra duality revisited: Paramonotonicity and operator splitting. *Journal of Approximation Theory*, 164(8):1065–1084, 2012.

[5] H.H. Bauschke and P.L. Combettes. *Convex Analysis and Montone Operator Theory in Hilbert Spaces*. Springer Science+Business Media, 2011.

[6] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing*, 18(11):2419–2434, 2009.

[7] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[8] A. Ben-Isreal and T.N.E. Greville. *Generalized Inverses*. Springer New York, 2003.

[9] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[10] B. Bethke, J.P. How, and A.E. Ozdaglar. Approximate dynamic programming using support vector regression. In *CDC*, pages 3811–3816. IEEE, 2008.

[11] J. Bioucas-Dias and M. Figueiredo. A New TwIST: Two-Step Iterative Shrinkage/Thresholding Algorithms for Image Restoration. *IEEE Transactions on Image Processing*, 16(12):2992–3004, 2007.

[12] R.I. Boţ. *Conjugate Duality in Convex Optimization*. Springer-Verlag Berlin Heidelberg, 2010.

[13] R.I. Boţ, E.R. Csetnek, and A. Heinrich. A primal-dual splitting algorithm for finding zeros of sums of maximally monotone opertators. Optimization Online, `http://www.optimization-online.org/DB_FILE/2012/06/3514.pdf`, 2012.

[14] R.I. Boţ and R. Csetnek. A comparison of some recent regularity conditions for Fenchel duality. In H.H. Bauschke, R.S. Burachik, P.L. Combettes, V. Elser, D.R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms of Inverse Problems in Science and Engineering*. Springer Verlag New York, 2011.

[15] R.I. Boţ, R. Csetnek, and G. Wanka. Regularity conditions via quasi-relative interior in convex programming. *SIAM Journal on Optimization*, 19(1):217–233, 2008.

[16] R.I. Boţ, S.-M. Grad, and G. Wanka. A new constraint qualification and conjugate duality for composed convex optimization problems. *Journal of Optimization Theory and Applications*, 135(2):241–255, 2007.

[17] R.I. Boţ, S.-M. Grad, and G. Wanka. *Duality in Vector Optimization*. Springer-Verlag Berlin-Heidelberg, 2009.

[18] R.I. Boţ, S.-M. Grad, and G. Wanka. New regularity conditions for Lagrange and Fenchel-Lagrange duality in infinite dimensional spaces. *Mathematical Inequalities & Applications*, 12(1):171–189, 2009.

[19] R.I. Boţ and A. Heinrich. Regression tasks in machine learning via fenchel duality. To appear in *Annals of Operations Research*, 2012.

[20] R.I. Boţ, A. Heinrich, and G. Wanka. Employing different loss functions for the classification of images via supervised learning. To appear in *Journal of Global Optimization*, 2012.

[21] R.I. Boţ and C. Hendrich. A double smoothing technique for solving unconstrained nondifferentiable convex optimization problems. arXiv:1203.2070v1 [math.OC], 2012.

[22] R.I. Boţ and C. Hendrich. On the acceleration of the double smoothing technique for unconstrained convex optimization problems. arXiv:1205.0721v1 [math.OC], 2012.

[23] R.I. Boţ and L. Lorenz. Optimization problems in statistical learning: Duality and optimality conditions. *European Journal of Operational Research*, 213(2):395–404, 2011.

[24] R.I. Boţ, N. Lorenz, and G. Wanka. Optimality conditions for portfolio optimization problems with convex deviation measures as objective functions. *Taiwanese Journal of Mathematics*, 13(2A):515–533, 2009.

[25] J.F. Bonnans, J.C. Gilbert, C. Lemaréchal, and C.A. Sagastizábal. *Numerical Optimization*. Springer Verlag Berlin Heidelberg, 2006.

[26] L. Bottou and C-J. Lin. Support vector machine solvers. In L. Bottou, Chapelle O., D. DeCoste, and J. Weston, editors, *Large-Scale Kernel Machines*, pages 1–27. MIT Press, 2007.

[27] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[28] A. Chambolle and T. Pock. A first-order primal-dual splitting algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[29] G. Chantas, N. Galatsanos, A. Likas, and M. Saunders. Variational bayesian image restoration based on a product of t-distributions image prior. *IEEE Transactions on Image Processing*, 17(10):1795–1805, 2008.

[30] O. Chapelle, P. Haffner, and V.N. Vapnik. Support vector machines for histogramm-based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055–1064, 1999.

[31] O. Chapelle, V. Vapnik, and J. Weston. Transductive inference for estimating values of functions, 1999.

[32] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

[33] D. den Hertog. *Interior Point Approach to Linear, Quadratic and Convex Programming*. Kluwer Academic Publishers, 1994.

[34] O. Devolder, F. Glineur, and Y. Nesterov. A double smoothing technique for constrained convex optimization problems and applications to optimal control. Optimization Online, `http://www.optimization-online.org/DB_FILE/2011/01/2896.pdf`, 2010.

[35] O. Devolder, F. Glineur, and Y. Nesterov. Double smoothing technique for infinte-dimensional optimization problems with applications to optimal control. CORE Discussion Paper, `http://www.uclouvain.be/cps/ucl/doc/core/documents/coredp2010_34.web.pdf`, 2010.

[36] O. Devolder, F. Glineur, and Y. Nesterov. Double smoothing technique for large-scale linearly constrained convex optimization. *SIAM Journal on Optimization*, 22(2):702–727, 2012.

[37] I. Ekeland and R. Temam. *Convex Analysis and Variational Problems*. North-Holland Publishing Company, Amsterdam, 1976.

[38] C. Geiger and C. Kanzow. *Theorie und Numerik restringierter Optimierungsaufgaben*. Springer-Verlag, Berlin Heidelberg New York, 2002.

[39] L. Göhler and G. Wanka. Duality for portfolio optimization with short sales. *Mathematical Methods of Operations Research*, 53(2):247–263, 2001.

[40] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis*. Springer-Verlag Berlin Heidelberg, 2001.

[41] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge Univerity Press, 1985.

[42] T-M. Huang, V. Kecman, and I. Kopriva. *Kernel Based Algorithms for Mining Huge Data sets: Supervised, Semi-supervised and Unsupervised Learning*. Springer Verlag Berlin Heidelberg New York, 2006.

[43] P.J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistic*, 35(1):73–101, 1964.

[44] S. Jayaraman, S. Esakkirajan, and T. Veerakumar. *Digital Image Processing*. Tata McGraw Hill Education Private Limited, 2009.

[45] T. Joachims. *Learning to Classify Text using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers, Boston Dordrecht London, 2002.

[46] V. Kecman. Support vector machines – an introduciton. In L. Wang, editor, *Support Vector Machines: Theory and Applications*, pages 1–47. Springer Berlin Heidelberg New York, 2005.

[47] K. Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2):307–319, 2003.

[48] T.N. Lal, O. Chapelle, and B. Schölkopf. Combining a filter method with svms. In I. Guyon, S. Gunn, M. Nikravesh, and L.A. Zadeh, editors, *Feature Extraction: Foundations and Applications*, pages 439–445. Springer-Verlag, Berlin Heidelberg, 2006.

[49] S. Mahadevan. Learning representation and rontrol in Markov decision processes: New frontiers. *Foundations and Trends® in Machine Learning*, 1(4):403–565, 2008.

[50] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

[51] Y. Nesterov. Excessive gap technique in nonsmooth convex optimization. *SIAM Journal of Optimization*, 16(1):235–249, 2005.

[52] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[53] Y. Nesterov. Smoothing technique and its applications in semidefinite optimization. *Mathematical Programming*, 110(2):245–259, 2005.

[54] W.S. Noble. Support vector machine application in computational biology. In B. Schölkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel Methods in Computational Biology*, pages 71–92. MIT Press, 2004.

[55] T. Pennanen. Dualization of generalized equations of maximal monotone type. *SIAM Journal on Optimization*, 10(3):809–835, 2000.

[56] R.C. Puetter, T.R. Gosnell, and A. Yahil. Digital image reconstruction: Deblurring and denoising. *Annual Review of Astronomy and Astrophysics*, 43(1):139–194, 2005.

[57] R.M. Rifkin and R.A. Lipper. Value regularization and Fenchel duality. *Journal of Machine Learning Research*, 8(Mar):441–479, 2007.

[58] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[59] M. Ruschhaupt, W. Huber, A. Poustka, and U. Mansmann. A compendium to ensure computational reproducibility in high-dimensional classification tasks. *Statistical Applications in Genetics and Molecular Biology*, 3(7), 2004. Article 37.

[60] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In J.W. Shavlik, editor, *Proceedings of the 15th International Conference on Machine Learning*, pages 515–521. Morgan Kaufmann, 1998.

[61] B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson. Shrinking the tube: A new support vector regression algorithm. In M.S. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processing Systems 11*, pages 330–336. MIT Press, Cambridge, 1999.

[62] B. Schölkopf, R. Herbrich, and A.J. Smola. A generalized representer theorem. In *In Proceedings of the Annual Conference on Computational Learning Theory*, pages 416–426, 2001.

[63] B. Schölkopf and A.J. Smola. *Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond.* The MIT Press, 2002.

[64] J. Shawe-Taylor and Christianini N. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[65] J. Si, A.G. Barto, W.B. Powell, and D. Wunsch. *Handbook of Learning and Approximate Dynamic Programming (IEEE Press Series on Computational Intelligence).* Wiley-IEEE Press, 2004.

[66] M.O. Stitson, A. Gammerman, V. Vapnik, V. Vovk, C. Watkins, J. Weston, and Tw Ex. Surrey. Support vector regression with ANOVA decomposition kernels. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, pages 285–292. MIT Press, Cambridge, 1997.

[67] A.N. Tikhonov and V.Y. Arsenin. *Solutions of Ill-posed Problems.* W.H. Winston, Washington, D.C., 1977.

[68] T. Van Gestel, B. Baesens, J. Garcia, and P. Van Dijcke. A support vector machine approach to credit scoring. *Bank en Financiewezen*, 2:73–82, 2003.

[69] V. Vapnik. *Estimation of Dependences Based on Empirical Data.* Springer Science+Business Media, 2006.

[70] V.N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, 1995.

[71] V.N. Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.

[72] S. Varma and R. Simon. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7, 2006. Article 91.

[73] J-P. Vert, K. Tsuda, and B. Schölkopf. A primer on kernel methods. In B. Schölkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel Methods in Computational Biology*, pages 35–70. MIT Press, 2004.

[74] G. Wahba and G.S. Kimeldorf. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95, 1971.

[75] G. Wanka. Multiobjective duality for the Markowitz portfolio optimization problem. *Control and Cybernetics*, 28(4):691–702, 1999.

[76] Y. Zhong, C. Zhou, L. Huang, Y. Wang, and B. Yang. Support vector regression for prediction of housing values. In *Proceedings of the International Conference on Computational Intelligence and Security CIS'09*, pages 61–65, 2009.

[77] C. Zălinescu. *Convex analysis in general vector spaces*. World Scientific, River Edge, 2002.

Lebenslauf

Persönliche Daten

Name:	André Heinrich
Adresse:	Schloßteichstraße 9 09113 Chemnitz
Geburtsdatum:	09. Juni 1980
Geburtsort:	Rochlitz

Ausbildung

1992–1999	Gymnasium Penig
1999–2000	Grundwehrdienst: 2./Pionierbatallion Gera
10/2006	Studium der Wirtschaftsmathematik an der TU Chemnitz
2002	Vordiplom Wirtschaftsmathematik
09/2005–11/2005	Praktikum bei Watson Wyatt Insurance Consulting GmbH, München
2006	Diplomarbeit in Kooperation mit der Ingenieurgesellschaft Auto und Verkehr GmbH
11/2006	Abschluss des Studiums der Wirtschaftsmathematik
04/2007–09/2009	Wiss. Hilfskraft/Wiss. Mitarbeiter an der Technischen Universität Chemnitz, Fakultät für Mathematik, Professur Approximationstheorie, Tätigkeit im Bereich Forschung und Lehre, u. a. Forschungsprojekt in Kooperation mit Continental Automotive GmbH, Limbach-Oberfrohna
seit 10/2009	Wiss. Mitarbeiter an der Technischen Universität Chemnitz, Fakultät für Mathematik, Professur Approximationstheorie, drittmittelfinanzierte Industriepromotion in Zusammenarbeit mit der prudsys AG, Chemnitz

Erklärung gemäß § 6 der Promotionsordnung

Hiermit erkläre ich an Eides Statt, dass ich die von mir eingereichte Arbeit "Fenchel duality-based algorithms for convex optimization problems with applications in machine learning and image restoration" selbstständig und nur unter Benutzung der in der Arbeit angegebenen Quellen und Hilfsmittel angefertigt habe.

Chemnitz, 28. September 2012

André Heinrich